Миграция из Windows в Linux с. 46,66,70

LXFDVD

ubuntu + openSUSE

ПЛЮС: Подшивка LXF за 2005–2007 год в PDF!

LINUX FORMAT

Главное в мире Linux



Январь 2008 № 1 (100/101)

«Open Source обладает мужеством и компетентностью: многие проиграют, победят лишь самые открытые.»

Майкл Микс

Новые горизонты

Курс на 2008 год: узнайте, что готовят нам Firefox, KDE и прочие

Рокировка с Windows
Засуньте Vista в виртуальную машину **с. 46**

Безупречный загрузчик
ПК не стартует? Узнайте, почему и почините! **с. 74**

IRC из Perl
Начните программировать сегодня **с. 82**

Займемся аппаратной частью
Соберите свой собственный гаджет и запрограммируйте его! **с. 54**

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882
Каталог «ПРЕССА РОССИИ» – подписной индекс 87974





Очумелые ручки

Вот интересно: если дать Команде LXF Arduino, пьезосирену, фотоэлемент и пару светодиодов, что у них из этого получится?



Пол Хадсон
iPhone – Apple взяла именно эти компоненты, пришпилила на них эмблему, а мужики-то и не знают!



Грэм Моррисон
Дискотека для грызунов – тоскующих по прошлому web взбодрит живая версия The Hamster Dance.



Майк Сондерс
Счастьеметр. Если фотодиод не засечет, что вы светитесь от радости, сирена мгновенно выдаст 190 дБ!



Эфрейн Эрнандес-Мендоса
Таймер, чтобы мои подданные точно знали, сколько им осталось... Ну, когда стану Властелином мира.



Мэтт Нейлон
Будильник, срабатывающий, как только Microsoft запатентует субъективность восприятия всего сущего. Ой, что это за звон?



Энди Ченнел
Панель управления для моего вечного двигателя. Жонглирование законами физики – лишь вопрос управления нулями и единицами.



Энди Хадсон
Лампочки, как на доспехах Дарта Вейдера – самое крутое, что я когда-либо видел. Сообщу миру, что я еще живой!



Д-р Крис Браун
Если добавите пару выключателей, я бы собрал Иванушку-дурачка, а потом и робокуклу Джорджа Буша.



Нейл Ботвик
Система мониторинга, следящая, чтобы никто в доме не использовал больше двух кусков туалетной бумаги. Я – борец за окружающую среду!



Дэвид Картрайт
Зонд для холодильника – уж он-то окончательно подтвердит, что свет выключается, когда вы закрываете дверку.

Миссия журнала



- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообщество
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО



Новый формат Linux Format

Не буду долго ходить вокруг да около – вы снова держите в руках юбилейный (сотый по оригинальной нумерации) и одновременно новогодний выпуск **LXF**. Двойной праздник, на который я намекал еще в LXF96, наконец-то состоялся!

Ну а какой же праздник без подарка? Посовещавшись в кулуарах, мы отбросили очевидно бесперспективные идеи вроде «каждому читателю по EeePC, каждой читательнице – по Linutop» и выбрали поистине бесценную вещь – себя, любимых. На DVD этого номера вы найдете подшивку всех номеров журнала за 2005–2007 год в формате PDF.

Но и это еще не все! Мы мечтаем рассказать о Linux как можно большему числу людей, поэтому, одновременно с выходом данного номера, **LXF** анонсирует новый проект – wiki.linuxformat.ru. На данном сайте планируется собрать и поддерживать (силами сообщества) в актуальном состоянии все доступные материалы LXF в вики-разметке.

Чтобы эта затея действительно удалась, нам необходима ваша помощь. Все статьи уже есть на DVD – просто сообщите нам, какие из них вы хотели бы перевести в вики. Мы создадим для вас учетную запись и вы сможете трудиться на благо всего русскоязычного Linux-сообщества. Подключайтесь! **LXF**

Валентин Синицын » Главный редактор info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru

› **Адрес редакции:** Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54.
› **Телефон редакции:** (812) 717-00-37. Дополнительная информация на стр.134

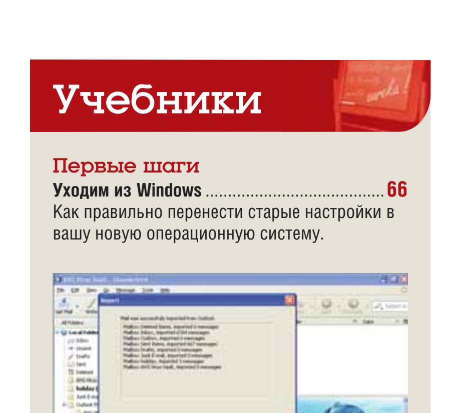


Январь 2008 № 1 (100/101)

Содержание

Весь номер – прямо как на ладони: приятного чтения!

Учебники



› Windows? В LXF? Мама...



LXF DVD100/101

Ubuntu 7.10
Запустите Linux прямо с DVD! Мы предлагаем вашему вниманию традиционно улучшенную версию, включающую KDE, Xfce, средства разработки и многое другое. Наслаждайтесь!

OpenSUSE 10.3
Последняя версия GNOME и предварительный релиз KDE4: да. Крутая новая тема: да. Плюс: тысячи других пакетов для этого мощного дистрибутива

Архив Linux Format
Все когда-либо выходившие на русском языке номера журнала, кроме текущего – теперь на диске и в PDF.



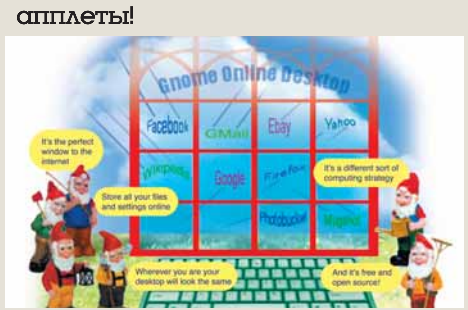
› Canonical доставит CD с Ubuntu в любую точку мира.



Что за штука... с. 36

Gnome Online Desktop?

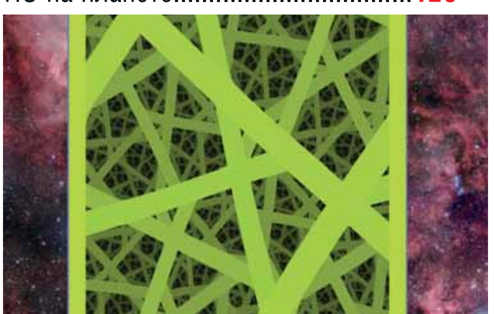
Прощайте, старые добрые апплеты!







LXF HotPicks



› Project M: обалденные аудиовизуальные эффекты для Linux.

Конструируем с Arduino с. 54

Превращаем маленькую, нашпигованную электроникой плату в эксклюзивный гаджет.





Подпишись на Linux Format и сэкономь!

Содержание

LXF DVD внутри!

Хотите узнать, что на нем? См. с. 126

Спецрепортаж

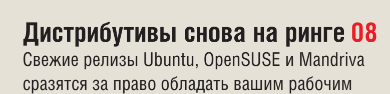
Новые горизонты



А также...

Постоянные рубрики

Обзоры





› Создавайте новые карты и играйте против других онлайн – свободно и задаром!

Сравнение: расширения Firefox

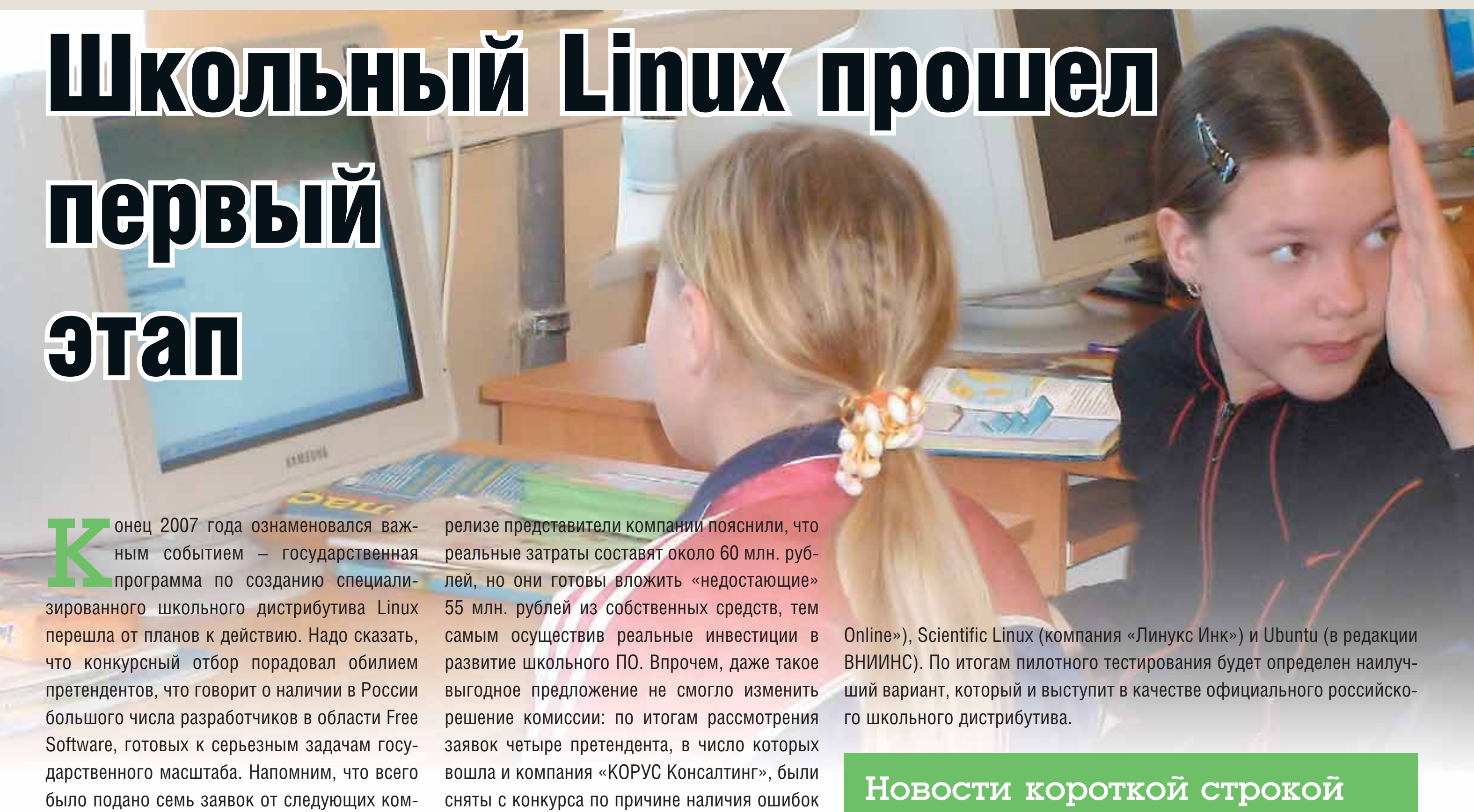
LINUX FORMAT Новости

» Рубрику ведет Илья Шпаньков



ГЛАВНЫЕ НОВОСТИ: » Школьный Linux – от слов к делу » «Компьютер в каждый дом» под прицелом ФАС » CuneiForm по открытой лицензии » Свободное ПО для образования, науки и культуры » Opera против Microsoft » Trolltech делает еще один шаг навстречу KDE » Российские суперкомпьютеры

Школьный Linux прошел первый этап



Конец 2007 года ознаменовался важным событием – государственная программа по созданию специализированного школьного дистрибутива Linux перешла от планов к действию. Надо сказать, что конкурсный отбор порадовал обилием претендентов, что говорит о наличии в России большого числа разработчиков в области Free Software, готовых к серьезным задачам государственного масштаба. Напомним, что всего было подано семь заявок от следующих компаний и организаций:

1. Компания «КОРУС Консалтинг» – заявленная стоимость проекта 5 млн. рублей;
2. Томский государственный педагогический университет – заявленная стоимость проекта 70 млн. рублей;
3. ЗАО «АйТи. Информационные технологии» – заявленная стоимость проекта 65 млн. рублей;
4. ООО «Институт информационных технологий» – заявленная стоимость проекта 40 млн. рублей;
5. ООО «РБК-Центр» – заявленная стоимость проекта 59,5 млн. рублей;
6. ЗАО «Ланит» – заявленная стоимость проекта 63 млн. рублей;
7. ЗАО «Вериселл Проекты» – заявленная стоимость проекта 66 млн. 495 тысяч рублей;

Особый интерес наблюдателей вызвал первый претендент – компания «КОРУС Консалтинг», предложившая наименьшую стоимость проекта. В официальном пресс-релизе представители компании пояснили, что реальные затраты составят около 60 млн. рублей, но они готовы вложить «недостающие» 55 млн. рублей из собственных средств, тем самым осуществив реальные инвестиции в развитие школьного ПО. Впрочем, даже такое выгодное предложение не смогло изменить решение комиссии: по итогам рассмотрения заявок четыре претендента, в число которых вошла и компания «КОРУС Консалтинг», были сняты с конкурса по причине наличия ошибок в представленной документации. Таким образом, в финале оказались три команды:

1. Томский государственный педагогический университет
2. ООО «РБК-Центр»
3. ЗАО «Ланит»

После тщательного изучения всех предложений победителем была объявлена компания «РБК-Центр», на плечи которой и ляжет основная тяжесть разработки и внедрения в пилотных регионах специализированного дистрибутива. И слово «основная» здесь фигурирует не случайно: дело в том, что собственно «РБК-Центр» займется вопросами внедрения, а для создания программного обеспечения в качестве субподрядчиков будут выступать разработчики дистрибутивов: ALT Linux, Linux-Online, Linux-Ink, ВНИИНС, работающие каждый над своим вариантом. Таким образом, школы в пилотных регионах получат сразу несколько операционных систем, включающих ALT Linux, Linux XP (компания «Linux-Online»), Scientific Linux (компания «Линукс Инк») и Ubuntu (в редакции ВНИИНС). По итогам пилотного тестирования будет определен наилучший вариант, который и выступит в качестве официального российского школьного дистрибутива.

Новости короткой строкой

- Нью-Йоркская фондовая биржа перевела свои серверы на использование операционной системы Linux.
- Компания Likewise Software выпустила под лицензией GPLv3 пакет Likewise Open, позволяющий упростить подключение Linux-машин к домену Active Directory.
- Компания Microsoft подтвердила свой интерес к проекту OLPC и сообщила, что специализированная версия Windows XP будет готова к тестированию на ноутбуке XO, предположительно, в январе 2008 года.
- Организация Linux Phone Standards Forum (LiPS) подготовила первую редакцию стандартов для мобильных устройств, использующих Linux.
- Вышла первая версия среды разработки Red Hat Developer Studio, выполненной в виде расширения для Eclipse.
- Компания Инфра-Ресурс объявила о выходе новой версии пакета офисных приложений *OpenOffice.org 2.3.1 Pro*.
- Компания Canonical Ltd. подготовила первый релиз системы управления версиями Bazaar 1.0.

Европа жаждет открытых стандартов



Норвежская компания Opera Software, которая вот уже на протяжении 13 лет выпускает популярный web-браузер *Opera*, подала в Европейскую комиссию жалобу, в которой обвинила компанию Microsoft в нечестной конкуренции на рынке браузеров и в несоблюдении web-стандартов W3C. По мнению норвежцев, редмондский гигант должен или исключить из состава операционной системы браузер *Internet Explorer*, или включить в дистрибутив альтернативные браузеры. Одновременно авторы Opera просят Европейскую комиссию обязать Microsoft обеспечить полноценную поддержку открытых web-стандартов в своем браузере. Дело в том, что общепринятые стандарты монополист реализует по-своему, добавляя различные проприетарные расширения, а в результате пользователи альтернативных браузеров нередко сталкиваются с тем, что оптимизированные для работы с *Internet Explorer* web-сайты некорректно отображаются в других аналогичных программах.

Одной из причин данного заявления норвежцев послужили недавние решения Европейской комиссии по вопросу исключения из состава Windows медиа-плейера *Windows Media Player* в целях обеспечения честной конкуренции для всех разработчиков аналогичных приложений.

Спустя некоторое время к инициативе Opera Software присоединился и европейский филиал Free Software Foundation. Президент FSFE Георг Греве [Georg Greve] пояснил, что хоть Opera Software и не является активным разработчиком свободного ПО, но в вопросах поддержки открытых стандартов их позиция во многом совпадает с мнением FSFE. Монопольное положение Microsoft мешает развитию и свободных браузеров, включая *Firefox* и *Konqueror*, поэтому руководство FSFE решило поддержать жалобу Opera Software. Следует отметить, что свои слова представители Фонда FSFE поддержали и делом, отправив в Европейскую комиссию соответствующее письмо.

Свободное программное обеспечение – в массы

МОО ВПП ЮНЕСКО «Информация для всех» представил очередную разработку в рамках проекта «Свободное ПО для образования, науки и культуры». На этот раз в набор входит не только свободное программное обеспечение, но и полноценная операционная система Mandriva Linux 2008.0 One. Данный комплект будет распространяться в бюджетные учреждения сферы образования, науки и культуры, а также в некоммерческие организации. Следует также отметить, что выпуск этих дисков был профинансирован ФГУ «Росинформресурс» при Министерстве промышленности и энергетики в рамках проекта по переводу образовательных учреждений Ханты-Мансийского автономного округа – Югры на открытое программное обеспечение, реализуемого Комитетом по информационным ресурсам Администрации Губернатора ХМАО, поэтому в первую очередь комплект будет распространяться среди бюджетных учреждений сферы образования, науки и культуры ХМАО. В дальнейшем планируется распространять его через сеть центров социально значимой информации, действующих в рамках Программы ПЦПИ.

Кстати, предыдущий выпуск набора свободного ПО для среды Windows, выпущенный МОО ВПП ЮНЕСКО «Информация для всех» в рамках проекта «Свободное ПО для образования, науки и культуры», в декабре был бесплатно разослан по всем федеральным и областным библиотекам России. Ранее сборник был бесплатно распространен среди участников ряда профессиональных мероприятий для библиотечного сообщества России и СНГ, включая международную конференцию «Восток-запад в век информационно-телекоммуникационных технологий», семинары «Электронные издания и проблемы авторского права» и «Библиотеки и просвещение в области прав потребителей».






О птичках – в открытую

Сегодня трудно найти приложение, аналогов которому не существовало бы в среде Free Software, но подобные «белые пятна» в мире свободного ПО все же встречаются. К одному из них относились системы оптического (кстати, о птичках...) распознавания кириллического текста – несмотря на востребованность подобного программного обеспечения, разработки велись, в основном, для англоязычных потребителей. Впрочем, теперь данная проблема должна кануть в Лету: компания Cognitive Technologies объявила о своем намерении открыть к марту 2008 года исходные коды популярного инструмента распознавания текста *CuneiForm*. Пока же программа просто доступна для бесплатной загрузки с web-сайта разработчиков. В качестве модуля, данное ПО используется в пакете *CorelDraw!* и поддерживает распознавание более 20 языков, включая русский. *CuneiForm* может распознавать любые полиграфические, машинописные гарнитуры всех начертаний и шрифты, получаемые с принтеров, за исключением декоративных и рукописных. В систему встроены специальные алгоритмы для распознавания текста с матричного принтера, плохих ксерокопий, факсов и машинописи. Дополнительно планируется открыть в Интернете специализированный ресурс, с помощью которого авторы организуют бесплатное распознавание текста документов для всех желающих. В данный момент *CuneiForm* работает исключительно в среде Windows, но после публикации исходных текстов не исключено появление версии и для Linux. В качестве основной причины своих действий представители компании называют желание привлечь к совершенствованию программы сторонних разработчиков, тем самым ускорив создание удобного и нужного продукта.

http://www.cuneiform.ru/





Игра по правилам

Министерство информационных технологий и связи в последнее время все чаще упоминается в связи с программой перевода российских школ на использование ОС Linux, всеобщей интернетизацией российского образования и прочими весьма полезными и своевременными проектами. Но не всегда благие начинания реализуются в соответствии с российскими законами, и пример тому – возбуждение Федеральной антимонопольной службой дела, связанного с программой «Компьютер в каждый дом», в отношении Мининформсвязи. По планам организаторов, данный проект должен решить проблему компьютеризации в сельских и отдаленных районах России, и к поставленным целям претензий у ФАС нет. Вопросы возникли к собственно реализации данной программы, а точнее – к условиям конкурса, проводимого в целях поиска наиболее подходящего поставщика вычислительной техники в регионы. Впервые о возможных нарушениях российского антимонопольного законодательства в проекте «Компьютер в каждый дом» заявил депутат Государственной думы Виктор Алкснис, подготовивший соответствующий запрос в Федеральную антимонопольную службу. Дело в том, что в условиях конкурса изначально фигурировала продукция только двух известных компаний – Microsoft и Intel. Первая обеспечивала поставки программного обеспечения, а вторая планировала укомплектовать каждый компьютер процессорами собственного производства. Таким образом, весь конкурс сводился к поиску поставщика, который должен был заняться изготовлением компьютеров с уже заранее указанными компонентами. Но современные реалии таковы, что и у операционной системы Windows, и у процессоров Intel на рынке имеются конкуренты, которым подобные условия конкурса просто не оставляли шанса поучаствовать в тендере. Именно этот факт и явился причиной очередного антимонопольного разбирательства. На момент написания данной статьи никаких официальных комментариев со стороны Министерства информационных технологий и связи не поступало.

Курс на сближение



Норвежская компания Trolltech решила объединить усилия с разработчиками графической среды KDE в целях достижения максимальной эффективности в подготовке программного обеспечения для мобильных устройств. Изменения взаимоотношений двух давних партнеров произошли как на техническом, так и на кадровом уровне. Прежде всего, в SVN проекта KDE перенесен разрабатываемый в Trolltech код, позволяющий обеспечивать кросс-платформенность для Phonon – платформы для мультимедийных приложений, являющейся частью KDE 4. Все добавления Trolltech будут распространяться под лицензией LGPL. Одновременно, один из авторов KHTML Ларс Кнолл [Lars Knoll] получил должность вице-президента по разработкам в Trolltech. В его обязанности теперь будет входить руководство созданием кросс-платформенных приложений и программ для мобильных устройств, работающих под управлением Linux. Таким образом, можно отметить желание Trolltech привлечь к созданию мобильного ПО на базе *Qt* большее число разработчиков свободного ПО, что позволит компании более активно покорять рынок программного обеспечения для данной сферы. Особое внимание планируется уделить дальнейшему совершенствованию движка для мобильных браузеров Webkit, завоевавшего определенную популярность среди пользователей мобильных устройств.

Суперкомпьютер для российского образования

Весной 2008 года в Научно-исследовательском вычислительном центре Московского государственного университета им. М.В.Ломоносова появится кластер с пиковой производительностью 60 Тфлопс, построенный совместно с Институтом программных систем РАН, российской компанией «Т-Платформы» и корпорацией Intel. Суперкомпьютер, получивший название «СКИФ», будет собран на базе 1250 четырехъядерных процессоров Intel Xeon и, по оценкам экспертов, к моменту запуска эта система станет самой мощной законченной суперкомпьютерной установкой в России и СНГ, а также позволит МГУ занять свое место в десятке самых мощных научно-образовательных вычислительных центров мира. Основу суперкомпьютера МГУ составят blade-системы собственной разработки компании «Т-Платформы».

Строительство нового вычислительного комплекса МГУ станет этапом реализации суперкомпьютерной программы «СКИФ-ГРИД» Союзного государства России и Белоруссии. Эта крупнейшая в мире установка на базе процессоров Intel нового поколения будет являться центральным элементом научной суперкомпьютерной сети «СКИФ Полигон», формируемой государственной программой «СКИФ-ГРИД». Весной 2008 года, вместе с новым кластером Южно-Уральского государственного университета, созданного на базе аналогичных blade-систем и суперкомпьютером «СКИФ Cyberia», распределенная вычислительная система «СКИФ Полигон» достигнет суммарной пиковой производительности в 88 Тфлопс. LXF

Пока верстался номер

Уфимский государственный авиационный технический университет (УГАТУ), компания АйТи, IBM и корпорация Intel объявили о запуске в УГАТУ собственного суперкомпьютера. Его основу составляют 266 blade-серверов IBM на базе 532 четырехъядерных процессоров Intel Xeon серии 5345. Расчетная пиковая производительность суперкомпьютера составляет 20 Тфлопс.

Работа над ошибками

В результате технической ошибки в репортаже с выставки FOSS & Linux Solutions, опубликованном в LXF99, не был указан автор фотографий – Александр Белкин.

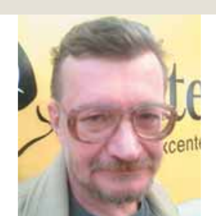
Редакция Linux Format приносит свои извинения

» При подготовке данного выпуска были использованы материалы сайтов www.armd.ru, rian.ru, www.cuneiform.ru, www.ifap.ru, www.linuxelectrons.com, www.trolltech.com, а также наших собственных корреспондентов.

LINUX FORMAT
Обзоры



Новинки программного и аппаратного обеспечения в описании наших экспертов



Алексей Федорчук
Его слабости – mass storage, разметка диска и файловые системы.

Поэзия – Linux'у

В предыдущей колонке я затронул тему Linux'а и творческой интеллигенции – чисто в академическом аспекте, мол, хорошо бы поэтам и писателям повнимательнее присмотреться к открытому ПО и альтернативным операционным системам. Но сегодня можно констатировать, что приобщение творческой интеллигенции к миру свободных ОС началось: 15 декабря 2007 года Linux был торжественно установлен на ноутбук Алисы Деевой, поэтессы, создателя литературного портала Фабула (http://fabulae.ru).

Я, конечно, знаю, что многие линуксоиды – поэты в душе, и среди них немало и таких, которые претворяют душевные порывы в строки и строфы. Но чтобы сложившийся поэт, далекий от IT-сферы, обратился бы к Linux'у – такого, как говорится, старожилы не припомнят.

В качестве дистрибутива был выбран Kubuntu 7.10 (i386) – ведь для пользователя-гуманитария наиболее подходящим будет самый гуманистический из всех Linux'ов. И практика подтвердила правильность выбора: несмотря на препоны и рогатки, обусловленные особенностями локального провайдерства, система была благополучно установлена и поступила в эксплуатацию. В результате на Руси появился первый поэт-линуксоид.

Событие это совпало по времени с другим, не менее знаменательным: юбилейным 100-м номером LXF. Правда, в английской его версии – но будем верить, что и русский 100-й журнал выйдет в свет в положенное время. А на его страницах мы будем видеть не только новости, обзоры, руководства, но и литературные произведения, написанные пользователями свободного ПО, с помощью свободного инструментария, и посвященные свободным сюжетам.

alv@posix.ru

Сегодня мы рассматриваем...

Дистробой с. 08



› OpenSUSE 10.3 и Ubuntu 7.10 можно найти на LXFDVD этого месяца!

Linutop с. 11



› Собираете терминал для использования в общественных местах? Засуньте внутрь такую штуку. Или просто покажите свою крутизну, поставив ее на рабочий стол!

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.



Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth
Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряющие практичная программа.*

Рейтинг 9/10

Матч Mandriva 2008 – OpenSUSE 10.3 – Ubuntu 7.10

Разглядывая три новые версии ведущих дистрибутивов Linux, **Нейл Ботвик** пытается угадать, какой вы предпочли бы найти под новогодней елкой.

Вкратце...

» Дистрибутив должен включать Linux-ядро, «неядерные» части ОС и набор разнообразного ПО. Мы сравнили три популярных дистрибутива.

Пока я пишу, в моей каминной трубе завывает ледяной октябрьский ветер, а магазины наполняются рождественскими подарками и безделушками. Linux-дистрибутивы, видимо, тоже принимают участие в акции: три из них выпустили новые версии в этом месяце, побудив нас к трехстороннему сравнению.

Mandriva 2008, OpenSUSE 10.3 и Ubuntu 7.10 вышли одновременно, и вместо отдельных обзоров нам показалось интереснее сравнить их, поставив рядом. К сожалению, финальный вариант Ubuntu 7.10 к нашему сроку не поспел, и мы воспользовались релиз-кандидатом. От финальной версии он отличается лишь возможным наличием недобитых ошибок, поэтому, если обнаруженный нами во время тестирования дефект числился в базе данных Ubuntu как исправленный, мы не упоминали его здесь, чтобы не искажать общей картины. Поиск и исправление ошибок в RC-версиях идут семимильными шагами: спустя четыре дня после выпуска уже насчитывалась почти сотня обновленных пакетов. Мы тестировали дистрибутивы как можно ближе к их стандартным конфигурациям. Там, где был выбор (например, какой рабочий стол установить), мы выбирали первый предложенный вариант.

> «На нашем жестком диске уже стояли Windows XP и PCLinuxOS.»



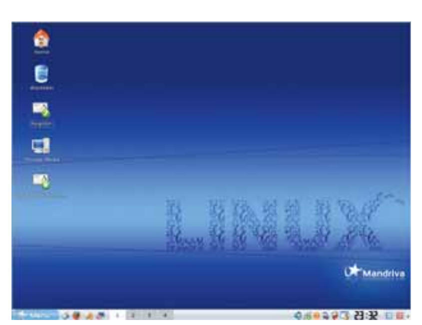
› **Установщик Mandriva предлагает KDE, GNOME и свободный выбор, когда можно ткнуть пальцем в конкретные пакеты.**

Установка

Mandriva предложила выбрать между стандартными KDE и GNOME и собственным набором пакетов. Первые два варианта переходят прямо к стадии установки, а третий предоставляет выбор пакетов как по отдельности, так и сгруппированных по категориям. Попытка установки на диск с Windows XP вызвала предложение использовать свободное место на разделе Windows, затем, после разумной подсказки о запуске *chkdsk*, размер раздела изменился, расчистив место для Mandriva. При установке на диск, где имеется не только Windows, но и другая Linux-система (в нашем случае, PCLinuxOS), Mandriva часто игнорирует собратьев, работая только с Windows-разделом и включая в загрузочное меню только Windows и себя: наша PCLinuxOS уцелела, но стала недоступна при загрузке. SUSE повел себя так же, а вот Ubuntu нашел все другие дистрибутивы и даже определил их названия, выделив им осмысленный пункт в загрузочном меню.

Mandriva теперь настаивает на установке пароля, да еще и приличной длины. Прощай, времена свободного входа от имени root! Пересмотрен подход к безопасности: ее уровень по умолчанию – High (Высокий), с включенным брандмауэром. Установка Mandriva прошла за несколько минут, значительно быстрее, чем в предыдущих версиях, и почти догнала Ubuntu. После перезагрузки система готова к работе, хотя есть и необязательное предложение зарегистрироваться в Mandriva, заодно приняв участие в анонимном опросе.

SUSE тоже предлагает выбор между KDE и GNOME, но кроме них – только текстовый и минимальный графический варианты. Выбор можно обогатить позже, в процессе установки, и возможна комбинация KDE и GNOME.



› **В стандартной теме Mandriva – рекордное число пингвинов, но они размещены со вкусом.**

SUSE предлагает даже бета-версию KDE 4 (наряду со стабильным KDE 3.5). Наличие вкладок Overview и Expert на разных стадиях процесса установки – мудрое решение: новичок не заблудится в дебрях системы, зато опытный пользователь получает полный контроль над процессом. Установка пакетов происходит раза в три медленнее, чем у Mandriva и Ubuntu, причем вопросы задаются и после установки системы. Установщик SUSE – единственный, который завершает свою работу уже в загруженном дистрибутиве, настраивая учетные записи пользователей и оборудование. *YaST* растерялся при настройке двух мониторов: характеристики взял от одного из них, широкоэкранного (1680 x 1050), а разрешение установил от второго, 1600 x 1200.

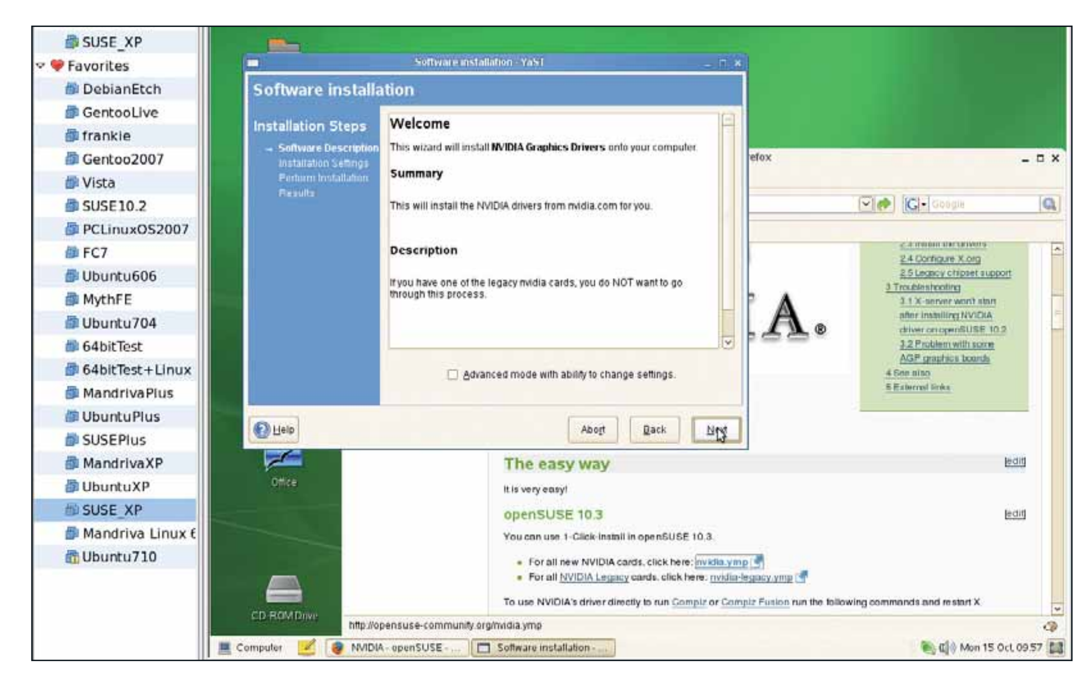
Установщик Ubuntu не только обнаружил Windows, но и предложил перенести оттуда пользовательские данные (например, интернет-закладки). Кроме того, в нашем тесте он единственный не только «разглядел» другую Linux-систему, но и создал для нее пункт в загрузочном меню.

Основной недостаток Ubuntu – он не дает контролировать процесс установки, хотя для

Версии ПО

	Mandriva	SUSE	Ubuntu
Ядро	2.6.22.9	2.6.22.5	2.6.22
XOrg	7.2	7.2	7.2
GNOME	2.20	2.20	2.20
KDE	3.5.7	3.5.7*	N/A
OpenOffice.org	2.2.1	2.3.0	2.3.0
Compiz-Fusion	0.5.2	0.5.4	0.6.0

*SUSE предлагает также KDE 4 (3.93).



неопытных пользователей это скорее преимущество. Вдобавок это Live CD, а значит, устанавливается то, что есть на диске, без всякого выбора. Конечно, можно добавить дополнительные пакеты после загрузки системы (установщик настраивает несколько репозиториев), но это куда менее удобно, чем установка пакетов с DVD, особенно при развертывании системы на нескольких компьютерах. С другой стороны, установка проходит гораздо быстрее, чем у SUSE – примерно так же, как у Mandriva. Ubuntu мог бы выиграть конкурс по простоте, если бы не манера устанавливать всю систему в единственный корневой раздел: все-таки отдельный домашний раздел сильно упрощает жизнь.

› Сказано: «это просто». И в самом деле, установка ПО с web-страницы «за один щелчок» в SUSE – способ простейший.

3D-рабочие столы

Установив дистрибутивы, мы первым делом затеяли проверку 3D-эффектов. Это же самые остросовременные дистрибутивы – пусть блеснут достижениями!

Проба Mandriva дала неожиданный промах – «Ваша система не поддерживает эффекты трехмерного рабочего стола», без всякого намека на причину: наш тестовый компьютер оснащен видеокартой Nvidia, а Mandriva Free (разумеется) не содержит двоичные драйверы. Так почему бы не дать пояснение или не предложить добавить репозиторий несвободного ПО (а ведь Mandriva Software Manager может сделать это легко и быстро)? После установки драйверов Nvidia осталось только выбрать, какой 3D-рабочий стол установить. Mandriva, как и другие дистрибутивы, предлагает *Compiz-Fusion*, с обычным набором дрожащих окон и вращающихся кубиков. Есть и менее эффектный, но более продуктивный *Metisse*, мы рассматривали его ранее в этом году.

SUSE тоже не смог настроить 3D, но он, по крайней мере, объяснил причину сбоя и подсказал способ добыть драйверы Nvidia. Бросок в Google обнаружил их установку «за один щелчок» на http://en.opensuse.org/NVIDIA, но пользователю приходится набрать несколько настроечных команд в оболочке. Включение этих команд в установочный скрипт могло бы превратить «однощелчковую» установку в истинное чудо.

А вот Ubuntu определил видеокарту Nvidia на ранней стадии, и выдал диалоговое окно с предложением установки проприетарных драйверов; он также, единственный из трех, справился с настройкой двух мониторов. Все, что для этого понадобилось – пара щелчков мыши да перезапуск X. Ubuntu серьезно повышает доступность Linux для технически неподготовленных новичков! Итак, 1 очко Ubuntu за автоматизацию процесса установки, 1 очко Mandriva за включение *Metisse* наряду с *Compiz*, и 1 очко SUSE за «однощелчковую» установку драйверов, которое мы, впрочем, забираем обратно за сокрытие опций на web-странице.

Тестовая среда

Для проверки аппаратной совместимости мы использовали все, что было под рукой, чтобы получить максимально достоверные результаты. Большинство других тестов проводилось на VMware Workstation, это позволяет полностью контролировать тестовую среду. Все дистрибутивы устанавливались на один диск с Windows XP, и все безошибочно «расчистили» для себя место. Тестирование 3D-эффектов проводилось на машине Core 2 Duo с видеокартой Nvidia 7100. Может быть, это уже не самая крутая конфигурация, зато она примерно соответствует среднему уровню парка личных компьютеров на сегодня, и более чем адекватна Compiz и Metisse.

Работа с мультимедиа

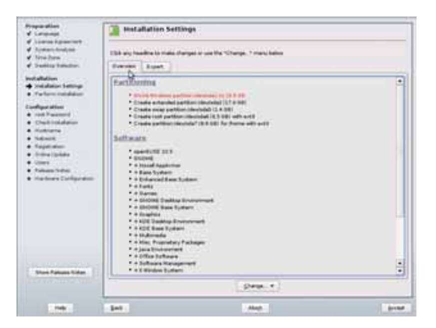
Воспроизведение DVD-видео, а иногда и MP3, ограничивается законами некоторых стран. Стало быть, вопрос не в том, могут ли дистрибутивы воспроизводить DVD (не могут), а в том, насколько сложно получить такую возможность. Penguin Liberation Front – давно известный неофициальный репозиторий Mandriva, и там есть все необходимое для воспроизведения DVD и файлов Windows Media, а для установки достаточно следовать инструкциям на http://easyurpmi.zarb.org.

Ubuntu показался было самым удобным: выбирайте пакеты «ограниченно свободного» ПО в утилите установки/удаления программ, и получите возможность воспроизведения MP3 и DVD вместе с необходимыми кодеками и Flash-модулем (сторонних программ *Automatix* и *Easy Ubuntu*, которые занимались этим в предыдущих версиях дистрибутива, больше нет). Однако *Totem* потребовал дополнительных файлов, например, модуль расширения MPEG2 *GStreamer*; почему бы не установить его вместе со всеми прочими, или не переключиться на *xine*-версию *Totem*, с которой проблем меньше? Словом, налицо явно лишняя работа.

SUSE тоже отказался проигрывать DVD, а *Kaffeine* выдал ссылку на страницу opensuse.org, где разъясняется ситуация и приводится руководство по добавлению репозитория Packman с необходимыми кодеками. В *YaST* есть соответствующий пункт, и информацию с web-страницы копировать незачем (ПО имеется на DVD, а установка с него намного проще), но и здесь пригодилась бы ссылка на установку «в один щелчок».

Добавление ПО

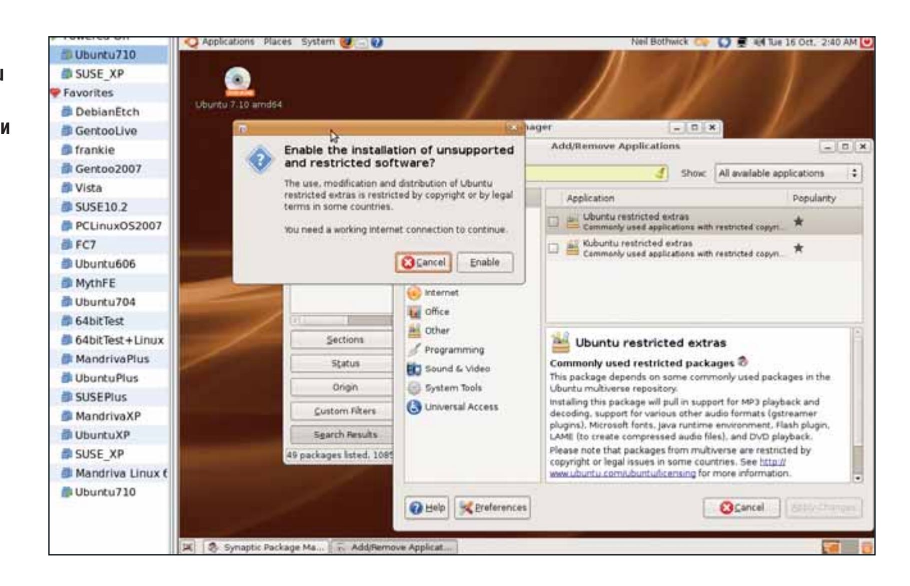
Если Ubuntu поработал над максимальным упрощением управления ПО в стандартной версии, то SUSE двигался в ином направлении, за две предыдущих версии совершенствуя свой менеджер пакетов (он был довольно неповоротлив). SUSE вернулся к единственному менеджеру пакетов, *YaST*, и он теперь прибавил в скорости. В 10.1 и 10.2 установка шла дольше, чем компиляция из исходников в Gentoo, а в 10.3 *YaST* работает отлично. Но подлинное достижение – «однощелчковая» система установки ПО. Теперь программу можно установить одним щелчком на ссылке web-страницы. »



› Установщик SUSE может принять большую часть решений за вас, а умникам позволяет разбираться самим.

» Пользователи Nokia N800 уже знакомы с подобной концепцией, как и пользователи Windows, которые щелчком по exe-файлу на домашней странице программы предоставляют программе загрузиться и установиться самой. Подход SUSE (и N800) несколько иной: добавлять репозитории в *YaST* по-прежнему необходимо, но «однощелчковый» установщик загружает и запускает скрипт, добавляющий репозиторий и устанавливающий ПО. Заодно вас будут извещать об обновлениях программ, коль скоро репозиторий входит в список доступных. Весьма возможно, что пока это лучший ответ на жалобы о трудности установки ПО на Linux, хотя пакеты остаются «привязанными» к единственному дистрибутиву.

Менеджер пакетов Mandriva всегда считался одним из лучших. Он по-прежнему хорош, но не особо изменился по сравнению с ранними версиями (хотя автоматизировано добавление некоторых репозиториев). Можно возразить, «зачем учить ученого»; тем не менее это показатель снижения темпов прогресса Mandriva. Данный дистрибутив меньше других отличается от своих предшественников. Правда, в нем отказались от мультяшной графики – кое-кто сочтет это крупным шагом вперед.



› Один пакет добавляет в Ubuntu много «несвободного», или ограниченного, ПО.



› Все три дистрибутива распознали Windows-раздел и установили двойную загрузку, но только Ubuntu сумел перенести из Windows пользовательские данные.

Итак, победил…

Все три дистрибутива справились с обычной аппаратурой, предложенной для установки, хотя Ubuntu сделал это в самой дружелюбной, предупредительной манере. Распознавание двух мониторов, работающих от одной видеокарты Nvidia – другая история, с ним вообще достойно справился лишь Ubuntu. SUSE запутался в мониторах (они немного отличались размерами), а Mandriva и до этого не дошел.

В целом, тестирование всей троицы вызвало легкое разочарование. Например, рабочие столы за последнюю пару лет развивались очень бурно: появились 3D-эффекты, поисковые машины и т.п. Сейчас их развитие ощутимо замедлилось, что в принципе неплохо: значит, технология дозрела до стабильности. Но застой не вызывает восхищения. Особенно это заметно в Mandriva, который выглядит прошлогодним дистрибутивом с подновленной темой, хотя настройка 3D хороша, а выбор из двух разных рабочих столов – еще лучше.

SUSE в прошлых версиях утратил ориентир, разбрасываясь между двумя менеджерами пакетов, каждый из которых был хуже одного прежнего. Похоже, что дистрибутив вновь нашел свой путь: SUSE 10.3 ухожен, менеджер ПО *YaST* скор и силен. А уж установку программ одним щелчком, по-моему, скоро переймут все дистрибутивы. Это реальная помощь в переходе с Windows на Linux, устраняющая одну из основных трудностей для новичка в незнакомой системе.

Ubuntu тоже постарался по возможности облегчить работу с Linux, и в общем преуспел. Хотя включить воспроизведение мультимедиа по-прежнему сложно, но легкость обращения с оборудованием (даже при проблеме двоичных драйверов) и простой интерфейс делают его отличным кандидатом на роль системы для дебюта. Пользователям SUSE придется несколько сложнее, зато перерасти дистрибутив им удастся не скоро. Mandriva где-то рядом, но «не дотянул». Может быть, в следующий раз… LXF



LINUX FORMAT Вердикт

Mandriva Linux 2008 Free

Разработчик: Mandriva
Сайт: www.mandriva.com
Цена: Свободная загрузка

Функциональность	8/10
Производительность	8/10
Простота использования	7/10
Документация	7/10

» *Mandriva сбавил обороты, но пока держится. Следующая версия покажет, что к чему.*

Рейтинг 7/10



LINUX FORMAT Вердикт

OpenSUSE 10.3

Разработчик: SUSE/Novell
Сайт: www.opensuse.org
Цена: Свободная загрузка

Функциональность	9/10
Производительность	8/10
Простота использования	8/10
Документация	8/10

» *SUSE укомплектован лучше других, и установка программ в один щелчок задумана хорошо, но YaST мог бы быть дружелюбнее к новичкам.*

Рейтинг 8/10



LINUX FORMAT Вердикт

Ubuntu 7.10

Разработчик: Ubuntu/Canonical
Сайт: www.ubuntu.com
Цена: Свободная загрузка

Функциональность	8/10
Производительность	8/10
Простота использования	9/10
Документация	8/10

» *Работа в Ubuntu удобнее всех прочих; это лучший выбор для пользователей начального и среднего уровней.*

Рейтинг 8/10

Linutop

Вероятно, это самая стильная из современных Linux-машин. Грэм Моррисон вставляет USB-накопитель и начинает искать, где же тут кнопка…

Вкратце...

» Linux в алюминиевом корпусе. Достаточно вставить USB-накопитель с дистрибутивом. Альтернативы: собранное самостоятельно устройство на базе платы EPIA или коммерческий Zonbu.

СПЕЦИФИКАЦИИ

Процессор: AMD Geode LX700 (x86)

ОЗУ: 256 МБ

Порты: 4 x USB 2.0, аудио ввод/вывод, VGA-вывод (SUB-D15), Ethernet (RJ-45)

Хранилище: Внешнее, 1-ГБ USB-накопитель прилагается

Размеры: 9,3 x 2,7 x 15 см

Вынимая Linutop из коробки, вы сразу видите: перед вами нечто необычное. Маленький алюминиевый корпус создает впечатление, что «это» нашли на дне воронки после катастрофы корабля инопланетян. Мало ли что покоится в грунтах около Тунгуски… На передней – судя по нанесенному логотипу – панели изделия красуются четыре прямоугольных гнезда, размером с USB-порт, а справа от них – два круглых, снабженных значками микрофона и наушников. На противоположной стороне изделия круглое гнездо для адаптера 9 В и сетевой разъем RJ45. Больше ничего. Нет даже кнопки включения: все управление питанием – воткнуть/выдернуть вилку. А само устройство – боец в весе пера, около 280 г.



› Со своими четырьмя USB-портами, гнездом питания, VGA, Ethernet и гнездами под микрофон и наушники, Linutop не сильно похож на обычный компьютер.

USB – или вон отсюда

При первом подключении всех проводов, стандартный BIOS устройства жалуется, что загрузиться ему неоткуда. Тут-то и пригодится USB-брелок с дистрибутивом Linux, поставляемый в комплекте с устройством. Воткните его в один из USB-портов. Еще два разъема пойдут под мышь и клавиатуру (ведь надо же как-то общаться с ОС); свободным останется только один – маловато на все внешние устройства, особенно если вам захочется использовать собственный USB-брелок для переноса данных.

Компьютер поставляется с переработанной версией Xubuntu, вариацией популярного дистрибутива Ubuntu с рабочей средой Xfce. Она включает видеодрайверы и даже совместима с несколькими DVB-устройствами USB для приема телевизионных передач. Благодаря поддержке USB 2.0 чтение/запись USB-накопителя сравнительно быстрые, поэтому время загрузки от включения питания до рабочего стола составляет 90 секунд – совсем неплохо для флэш-устройства. Модель процессора – AMD Geode LX 700, объем памяти 256 МБ, и этот набор справляется с Xfce без каких-либо проблем. Пользовательский интерфейс отзывчив, а основные приложения вроде *Firefox* или *AbiWord*, с редким единодушием запускаются за 10 секунд. Удивило включение *Gaim* в качестве клиента службы мгновенных сообщений, ведь он давно был замещен *Pidgin*. Имеется и *Skype*, хотя ничто не мешает добавить собственные приложения с помощью менеджера пакетов *Synaptic*. В общем, нельзя сказать, что существо под алюминиевой оболочкой настроено враждебно.

Дань «зеленым»

Linutop сверхэффективен: он потребляет всего 5 Вт. Он бесшумен и не нагревается – идеально для случаев, когда нужна постоянно включенная надежная машина. И он может принести немало пользы, помимо уважения компьютерщиков, увидевших его на вашем столе. Несколько таких приборов могут составить сеть «тонких» клиентов. Linutop вполне справится с ролью терминала по продажам в магазине или интернет-киоска. Вот только четырех USB-портов маловато, если и ввод-вывод, и хранение осуществляются только через них. А еще – время нужно устанавливать заново при каждой перезагрузке, а значит, понадобится некая сетевая система синхронизации времени. Неплохо было бы оснастить компьютер доступом к беспроводной сети, чтобы не тянуть к нему кабель. И кнопка питания не помешала бы. А в остальном, футуристическая машинка заслуживает всяческих похвал. LXF



LINUX FORMAT Вердикт

Linutop

Разработчик: Linutop
Сайт: www.linutop.com
Цена: 280 Евро без НДС

Функциональность	7/10
Производительность	8/10
Простота использования	8/10
Оправданность цены	9/10

» *Умелый ящичек, способный пригодиться и дома, и в вестибюле музея или отеля.*

Рейтинг 8/10

ASUS Eee PC 701

Ультра-микро-портативный PC на Linux за какие-то 200 фунтов? **Майк Сондерс** с трудом поверил, что это не сон…

Eee на видео
В разделе нашего **DVD Magazine/Video** вы можете увидеть малютку в действии!

Вкратце...

» Сверхтонкий Linux-ноутбук производства ASUS, с установленным Xandros. *См. также* OLPC XO-1.



Годами нам грезилось компактное Linux-устройство без излишеств. Конечно, SharpZaurus и Nokia N800 – приятные штучки, но их едва ли можно назвать идеалом для Linux-пользователя, из-за крохотной клавиатуры и проприетарного оборудования. Поэтому в июне, когда тайваньский технологический гигант ASUS объявил о выпуске 199-долларового Eee PC под Linux, у нас и слюнки потекли. Вот он, настоящий ноутбук: у него есть BIOS, процессор Intel, изобилие встроенных функций, и он также может работать под Windows.

Теперь, после месяцев томительного ожидания, он у нас в руках. И он компактный. Не «зачем-то по-дурацки мелкий», не «мелкий на грани пригодности», а просто – суперкомпактный. Eee, что означает Easy to Learn, Easy to Work, Easy to Play (Легко учиться, Легко работать, Легко играть), продвигается как дешевый второй компьютер, который можно даже отдать детям благодаря его прочности (вместо жесткого диска в нем используется флэш-накопитель).

Увы, ASUS не удержался в рамках $199, и данная версия продается в Америке за $399 (вариант Eee PC с «урезанными» ОЗУ и флэш-памятью готовится к выпуску). Рекомендованная розничная цена впечатляет, особенно на фоне других ультрапортативных устройств (не менее $2 000); по ценовым соображениям Eee можно поставить в одну категорию с полноценными ноутбуками от Dell и К°.

Статистика

» **Размеры:** 22,5 x 16,4 x 3,5 см
» **Вес:** 0,92 кг
» **Процессор:** 900 МГц Intel Celeron
» **Память:** 512 МБ ОЗУ
» **Флэш-накопитель:** 4 ГБ
» **Свободное место после стандартной установки:** 1,3 ГБ
» **Сеть:** 802.11b/g Atheros WiFi, Ethernet
» **Порты:** MMC/SD, VGA, 3 x USB, микрофон, наушники
» **Видео:** Графика Intel GMA 900 (8 МБ разделяемой памяти), 800x 400-пиксельный экран 7”, 0,3-мегапиксельная web-камера
» **Основное ПО:** ядро 2.6.12, *glibc 2.3.6*
» **Рабочий стол:** X.org 7.1, IceWM 1.2.30, KDE 3.4.2
» **ПО:** *OpenOffice.org 2.0.4*, *Firefox 2.0.0.7*, *MPlayer 1.0rc1*
» **Время загрузки:** 26 секунд до рабочего стола
» **Время выключения:** 9 секунд
» **Время запуска OpenOffice.org*:** 12 секунд
» **Время запуска Firefox*:** 3 секунды
» **Свободного ОЗУ после загрузки:** 320 МБ
» **Количество работающих процессов после загрузки:** 61

*Примечание: цифры отображают время «холодного» запуска – то есть загрузки из такого состояния, в котором приложения не были кэшированы в ОЗУ. Естественно, в ином случае запуск происходит быстрее (например, для *OpenOffice.org* – 6 секунд)

› **Рабочий стол Eee, благодаря «чуткому руководству» *IceWM*, быстр, красочен и отлично организован по категориям через вкладки.**

В коробке вы найдете сам Eee, зарядное устройство, проспект технических обновлений, стартовое руководство и учебник на DVD по обращению с предустановленным дистрибутивом Xandros Linux и восстановлению ОС до «заводского» состояния с помощью внешнего CD/DVD-привода или USB-брелка.

За такую цену трудно ожидать высокого качества сборки, но Eee PC на удивление хорош. Шарниры экрана крепкие, а корпус жесткий – конечно, не Panasonic Tough, но толчки средней силы выдержит. Батарея вставляется в паз между шарнирами. Наш пробный экземпляр – ярко-белый, как видно на фото; в дальнейшем ASUS планирует выпускать изделия черного и, возможно, некоторых других цветов.

Возьмиии...

23 см в ширину, 16 – в высоту и 3,5 в толщину: казалось бы, где здесь место для расширений? Но ASUS умудрился втиснуть сюда целый набор портов. Кроме трех USB, здесь есть Ethernet, VGA-выход и модемный порт, закрытый резиновой заглушкой (модема внутри нет). Для хранения данных в Eee есть слот для карт памяти MMC/SD; при том, что 4-ГБ карты SD стоят дешевле $40, удвоить накопительную способность ноутбука можно совсем недорого. Можно даже обновить RAM через съемную панель на днище компьютера. Привода CD/DVD нет: такая добавка лишила бы Eee компактности.

Дистрибутив Xandros, Linux для Eee (вместе с разделом восстановления), оставляет 1,3 ГБ свободного места для файлов при первой загрузке. Хотелось бы иметь побольше свободного места после инсталляции, но и этого в общем хватает для нескольких MP3 и фильма, а также офисных документов и картинок – обычный дорожный набор. В качестве оконного менеджера Xandros выбрал *IceWM*, сверхскоростное решение, оснащенное красочными значками запуска приложений, сгруппированными по категориям. Есть и строка состояния, показывающая уровень зарядки аккумулятора, статус сетевого подключения и пр.

Для выхода в Интернет через Ethernet или Atheros WiFi, Eee снабжен *Firefox* и *Thunderbird*, которые надежно работают, несмотря на ограничение дисплея 7” 800x480. Некоторые сайты «заточены» под экраны настольных компьютеров разрешением 1024x768 и требуют горизонтальной прокрутки, но мелкие шрифты позволяют выйти из положения. К счастью, поддержка Adobe Flash на месте, и YouTube работает отменно, и можно воспроизводить файлы MP3 и Windows Media – при отсутствии аллергии на поддержку медиаформатов Microsoft! Все это исправно «тянет» связка из

процессора 900 МГц и 512 МБ ОЗУ – может быть, она и не поладит со сверхвысококачественным видео, но файлы, извлеченные нами из дисков, отнюдь не спотыкались.

Среди других приложений – *OpenOffice.org 2.0* (староват, но есть надежда, что ASUS или Xandros в скором времени выпустят обновление), *Skype*, *MPlayer*, *Amarok* и малые KDE-утилиты (калькулятор, блокнот и т.п.); несколько игр. 3D-игры (гонка *Planet Penguin* и *Crack Attack*), хоть и запускаются, но изрядно перегружают чип Intel GMA 900: дети еще могут поиграть, но нас торможение кадров изрядно достало. Управление файлами осуществляется собственным менеджером Xandros, с помощью которого можно запустить терминал, просмотреть файловую систему или настроить ОС.

Нажмиии…

Само собой, микроскопический размер не мог не сказаться на выборе ПО. Это не машина для редактирования фото, кодирования видео или компилирования KDE из исходников. Если главный фактор выбора – цена, то лучше уж малость переплатить и взять Dell за $500-600 – он, правда, более увесистый, зато вы получаете большой экран и накопитель. Но для тех, кто гонится за портативностью, недостатки с лихвой искупаются: основные приложения, типа *Firefox*, *OOo* и *Kopete*, работают уверенно, не хуже, чем на крупной тяжелой машине. При не слишком суровых требованиях к малютке разочарованы вы не будете.

На сегодня доступно только несколько добавочных пакетов для загрузки из репозиториев ASUS, да несколько обновлений. Немало пользователей сообщают об успешном подключении репозиториев стабильного Debian и об установке дополнительных программ, например, *Gimp*. В нашем же тестировании пакеты Debian потребовали обновления коренного ПО (*glibc*). Для прожженных линуксоидов подобные мелочи не в диковинку, а вот новичок наверняка растеряется. Активное сообщество на www.eeeuser.com работает над созданием собственного репозитория ПО для Eee.



Пол считает…

«Я дал бы ему 7,5/10. Да, он мал и дешев, но экран не дорос до нужных мне сайтов, и во многих приложениях надо свернуть панель задач, чтобы увидеть весь интерфейс. Часть файлов справки отсутствует, добавить ПО не так-то просто, Wi-Fi после перезагрузки не цепляет, а очередное обновление ASUS изменило язык одной из программ на китайский! Мелочи, конечно, но в целом все это смотрится как недоделанное и незаконченное. Экран побольше сильно улучшил бы картину.»



И как же на Eee работается? Клавиатура упругая и на вид хрупкая, но не спасовала перед набором в стиле долбежки (одними указательными пальцами). У сенсорной панели вроде бы одна кнопка, но под ней еще имеются два переключателя – без правого щелчка не останетесь. Web-камера со скромным разрешением 0,3 мегапикселя более чем достаточна для онлайн-чата или записи простеньких фильмов. Во время тестирования температура устройства держалась на приемлемом уровне: об этом заботится крошечный вентилятор. Время жизни аккумулятора зависит, как у всех ноутбуков, от работы WiFi и аппетита запущенных программ: в одном из тестов мы работали с *OOo* и *Firefox* через WiFi, да еще при музыкальном сопровождении, и батареи хватило на 2 часа 40 минут. Весьма неплохо; а если устройством пользоваться для кодирования или набора текста, то оно протянет часа четыре.

Плюсы Eee: дешев, компактен, работает под Linux, в том числе на других дистрибутивах, а то и на Windows XP; надежный (нет винчестера), расширяемый: словом, одна из приятнейших вещиц, способных попасть вам в руки. Здесь есть BIOS. Есть USB-порты. Его можно перепрограммировать. Можно швырнуть его в сумку – и в пути заниматься программированием, просматривать Интернет, делать офисную работу или смотреть кино.

Поймиии…

Минусы? Мы изменили бы разве что набор ПО, взяв новую версию *OpenOffice.org*, *AbiWord* в качестве облегченной альтернативы и кое-какие дополнительные приложения из репозиториев ASUS. Инструменты настройки работают надежно, только вот поддержка iPod «хромает» – пока работа с ним возможна только в дисковом режиме (поэтому нет полной интеграции с *Amarok*). Но все это мелочи, легко устранимые обновлениями ПО от ASUS или от сообщества. В целом, устройство сказочное: уступая, по понятным причинам, более крупным ноутбукам своей ценовой категории, оно вполне на высоте при просмотре Интернета, наборе текста и воспроизведении медиа. Не спешите подозревать, что ASUS подкупил нас, подарив тестовое устройство: у нас был только один экземпляр (с возвратом) на все компьютерные журналы Future Publishing. Лучше позвольте мне привести следующий аргумент: я куплю себе ЭТО при первой возможности. Равно как Дейв и Майк из *PC Format* и Алекс из *PC Plus*. Наш Грэм тоже хочет такой, и даже Келли, наш издатель, была покорена утонченностью вещицы, узнав, что среди расцветок будет нежно-розовая. LXF



› Хотя некоторые сайты при разрешении 800x480 выглядят неважно, в основном малютка работает гладко.



LINUX FORMAT Вердикт

ASUS Eee PC 701

Разработчик: ASUS
Сайт: http://eeepc.asus.com/en/
Цена: $399,99

Функциональность	9/10
Производительность	8/10
Простота использования	10/10
Оправданность цены	10/10

» *Удивительная, востребованная, расширяемая – мини-чудо с Linux (или нет) внутри и более чем привлекательной ценой.*

Рейтинг 9/10

Fedora 8





Современный мир Linux полон здорового соперничества: **Энди Хадсон** считает, что Fedora 8 таки утрет нос Gutsy Gibbon…

Вкратце...

» Один из популярнейших Linux-дистрибутивов. См. также: OpenSUSE или Ubuntu.

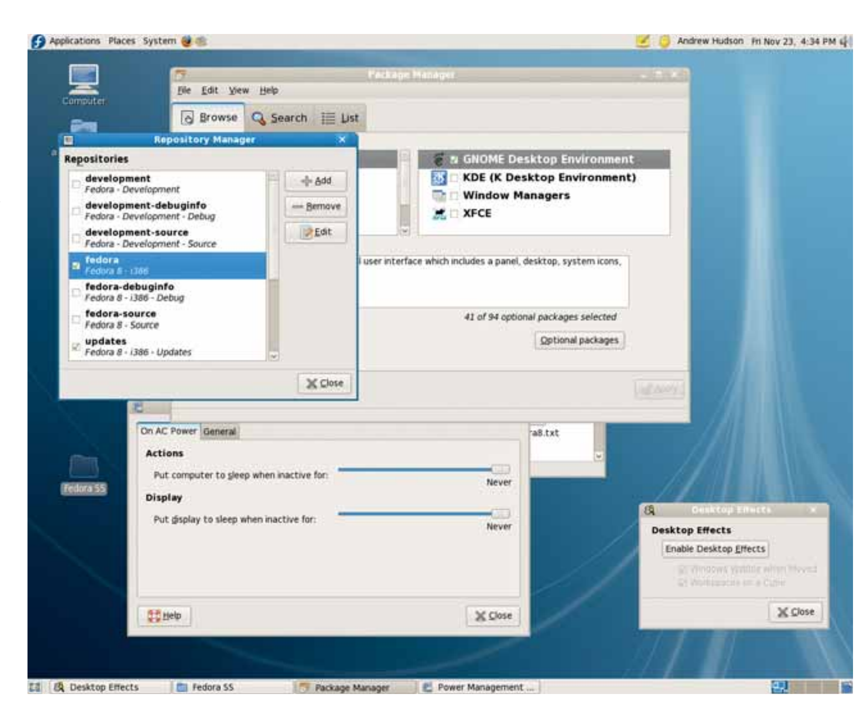
За последние 4 года проект Fedora продвинулся весьма заметно. Дистрибутив отпочковался от Red Hat Linux, когда в RH решили ускорить развитие за счет подключения к разработке участников сообщества. Это было тогда, а теперь с нами Fedora 8. В Азии восьмерка считается счастливой цифрой: повезет ли Fedora?

Красота

Исчезла унылая тема Echo, уступив место современной Nodoka. Новая тема смотрится профессионально и элегантно, особенно при работе *Compiz*, с его непременными дрожащими окнами и вращающимся кубиком. Визуально освежили даже инсталлятор *Anaconda*, отбросив воздушные шарики Fedora 7 и спирали Fedora 6 ради простой, но элегантной темы. Если вы любите внешний лоск, вам понравятся новые обои Fedora 8, поставляемые по умолчанию. Со сменой времени суток меняется и цвет оформления рабочего стола: утром и вечером заря занимается не только в небе, но и на экране. Gnome 2.20 тоже немало способствует общему великолепию Fedora, выводя дистрибутив в ряды наиболее современных.

Впрочем, известно: не графика «делает» Linux-дистрибутив. Так чем же еще отметилось сообщество Fedora со времени версии 7? Честно говоря, закрома не так уж богаты, но это не значит, что дополнения несущественны. Интеграция в Fedora *PulseAudio* обеспечила тонкий контроль громкости в пределах дистрибутива. Если одновременно обрабатывается несколько аудиопотоков, то уровень громкости каждого можно менять отдельно, а не на уровне ОС. Аудио- и видеоприложения Gnome тоже выиграли от интеграции *PulseAudio*, и RhythmBox теперь опознается по умолчанию. Вероятно, еще важнее, что команда Fedora, наконец, признала, что пользователь должен сам выбирать между проприетарными и свободными кодеками, и включила в дистрибутив *CodecBuddy/Codeina* для упрощения загрузки мультимедиа-кодеков (включая MP3). Интересно, что разработчики предпочли для MP3 расширение *Fluendo*, но при этом поставили пользователя в известность относительно разницы между проприетарным и свободным ПО.

А еще нам понравилась манера обращения Fedora с нашей web-камерой (Logitech QuickCam Pro 3000). В предыдущих версиях дистрибутивов ветки Red Hat она никогда не работала «по умолчанию», но с Fedora



› При улучшенной внешности Fedora, несомненно, завоюет немало новых поклонников.

Шаг за шагом: Получаем поддержку MP3 в Fedora



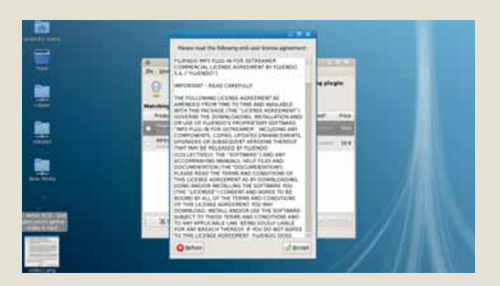
» Предупреждение

Если вы пробуете открыть неподдерживаемый медиа-файл, то Fedora сообщает о том, что его формат проприетарный, и предлагает свободную альтернативу.



» Выбор

Если вы ответили OK, то Fedora покажет возможные варианты открытия выбранного файла. Выберите то, что вам нужно, и нажмите Get Selected [Получить выбранное].



» Вы уверены?

Перед загрузкой кодека вам будет представлена лицензия. Неплохо будет хотя бы бегло прочесть ее – ведь это вам выбирать, принять или отклонить.

достаточно было просто подключить устройство к разъему, и уже через несколько минут мы, запустив *Ekiga*, участвовали в видеоконференции.

«Под капотом»

По умолчанию Fedora оснащается ядром 2.6.23, с предусмотренным переходом на 2.6.24, как только оно будет протестировано и выпущено. Улучшена поддержка виртуализации, и теперь, с Xen, KVM и QEMU, Fedora становится мощной виртуализационной платформой. Администраторам серверов будет интересно узнать, что виртуальные машины теперь можно администрировать удаленно, так что приходить в серверную уже необязательно. SELinux, естественно, тоже усовершенствован, работа с безопасным рабочим окружением стала проще и удобнее.

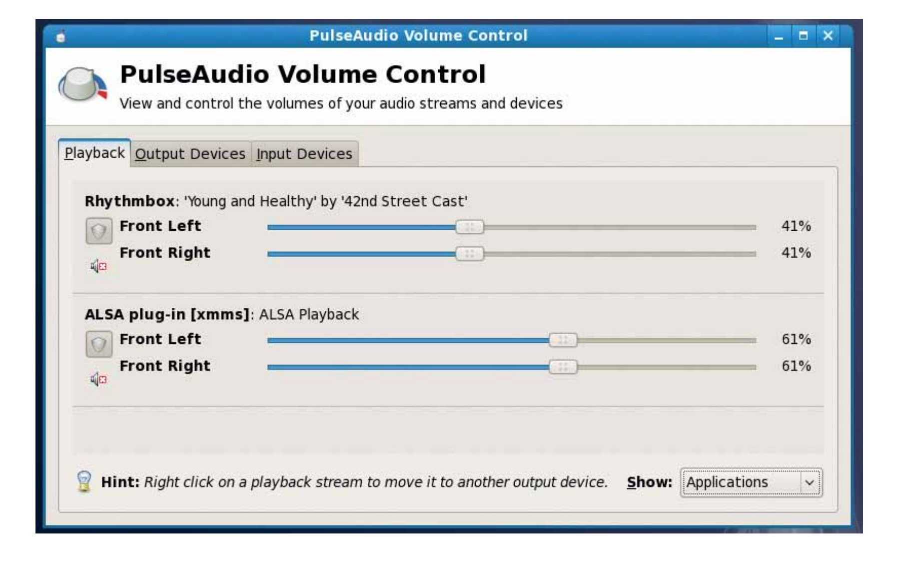
Очень важно для Fedora долгожданное улучшение управления пакетами. *Yum* заметно ускорился со времени Fedora 7, а *pirut* (стандартный графический инструмент управления пакетами), наконец, «научился» добавлять, редактировать и удалять репозитории и работать с CD/DVD. Хотя и проигрывая в скорости *apt*, *Yum* изрядно вырос, и, вкупе с дополнительной переработкой системы RPM, вскоре даст нам ощутить выгоды от этого роста.

Но это еще не все графические усовершенствования: утилита настройки брандмауэра (бывший бедный родственник семьи *system-config*) стала гораздо более гибким инструментом. А как лихо Fedora разделывается с принтерами! Пользователю надо лишь включить разъем устройства в порт, остальное система сделает сама. Появится уведомление об определении нового устройства, и секунд через тридцать вы найдете принтер в меню своего любимого приложения.

Если «копнуть» поглубже...

Указывает ли Fedora 8 на серьезные сдвиги в разработке дистрибутива? В прошлом график проекта был более чем гибким (помните девятимесячный цикл Fedora Core 5?), но теперь обещано придерживаться шестимесячного цикла – апрель и октябрь – отражающего график версий Ubuntu. Fedora 8 получил на разработку 5 месяцев, чтобы уложиться в новый цикл, так что идея Ubuntu о важном значении предсказуемости релизов явно взята в команде Fedora на вооружение.

Фактически, в Fedora мало что изменилось, просто эта версия добавила дистрибутиву зрелости, недостававшей предыдущим трем вариантам. Не поймите меня превратно, эти дистрибутивы, как таковые, не были плохи, но именно сейчас разработчики смогли, наконец, окончательно отшлифовать свое изделие. Несомненно, «приятность» теперь входит в число функций Fedora, и это привлечет новых поклонников.



» *PulseAudio* позволяет регулировать уровень громкости каждого аудиовыхода по отдельности.



Пол считает...

«Это первый дистрибутив Fedora за несколько лет, который меня действительно восхищает – мне нравится новый интерфейс брандмауэра, а оформление просто сказочное! Так держать, команда Fedora!»

Выбор оружия

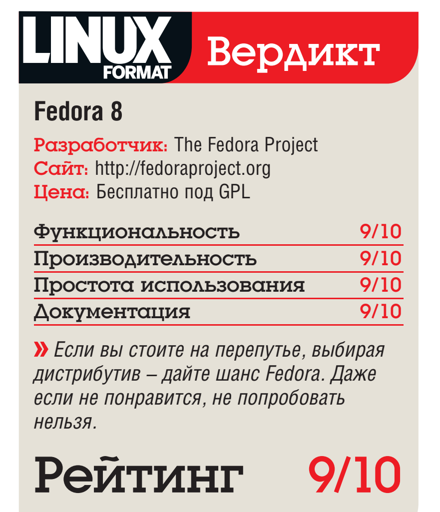
В рамках цикла Fedora 7 сообществу был представлен *revisor*, с помощью которого можно было создавать собственные версии Fedora, и эта идея получила дальнейшее развитие с выходом DVD-варианта, варианта GNOME Live CD, отдельных наборов KDE (3.5.8), Games и Electonic Lab, каждый из которых существенно расширил потенциальный круг пользователей Fedora. Простота создания собственной версии играет на руку дистрибутиву: вообразите, что вы запросто создаете особый вариант для своего офиса, который можно развернуть на 50-ти машинах вместе со специальным установочным скриптом *Kickstart*, и вы оцените всю подаренную вам мощь и удобство.

Так чего же нам ждать от проекта Fedora? Версия 8 дает понять, что вскоре последует жесткое соперничество с Ubuntu за количество пользователей. Древний спор о совместимости с MP3 после включения *CodecBuddy* потерял смысл; и появились мощные инструменты, предлагаемые Fedora для настройки своей системы. Рискнем даже предположить, что Fedora становится серьезным конкурентом Ubuntu по части простоты использования и дружелюбия. Если вам нужен гибкий дистрибутив, остающийся при этом на позициях свободного ПО, то Fedora должен занять в вашем списке одну из верхних строчек. Он резко улучшился, а *CodecBuddy* значительно облегчает его использование в качестве мультимедиа-платформы.

Судя по событиям прошлого, Fedora 9 должен послужить основанием RHEL 6, поэтому серьезных изменений между 8 и 9 ждать не стоит. Вместо этого дистрибутив ожидают дальнейшая шлифовка и доводка до мирового класса. Если вы еще не пробовали Fedora – самое время вливаться! LXF



» **Готово**
Fedora загрузил нужный кодек, установил его и приступил к фоновому воспроизведению файла. Все отлично в этом мире!



LINUX FORMAT Вердикт

Fedora 8

Разработчик: The Fedora Project
Сайт: http://fedoraproject.org
Цена: Бесплатно под GPL

Функциональность	9/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если вы стоите на перепутье, выбирая дистрибутив – дайте шанс Fedora. Даже если не понравится, не попробовать нельзя.*

Рейтинг 9/10

Alien Arena 2007

Не терпится нажать на спуск? **Грэм Моррисон**, кажется, нашел способ поддержать форму в ожидании очередной версии Unreal Tournament.

Вкратце...

» Стрелялка «от первого лица» на движке *Quake*. Чтобы добиться реальных успехов, играйте со знакомыми (или незнакомыми) партнерами. См. также: *Unreal Tournament* или *Terminus*.

Открытые компьютерные игры нечасто попадают к нам в Обзоры, но *Alien Arena 2007* – весьма необычный экземпляр. Прежде всего, она основана на ПО, ранее бывшем проприетарным. Исходный код культовых игр *Quake II* и *Quake III* был опубликован id Software под GPL несколько лет назад, они-то и составили сердцевину *Alien Arena*. Но это не значит, что разработчики не внесли в игру ничего нового. Объем работы, проделанной ими над исходным кодом, и количество дополнений к механике и арсеналу игры просто поражают. Здесь и движок частиц, и яркие цветовые эффекты, и текстуры высокого разрешения, и 42 уровня игры ручной работы, причем модели и игровой дизайн вполне им под стать. Программисты явно работали над *Alien Arena* с любовью, и их внимание к деталям обнаруживается сразу после запуска игры.



› **Бурно и неистово – вот главное ощущение от игры в *Alien Arena*. Это классическая стрелялка от первого лица, и размышлять тут некогда: слишком тесно.**

А что же сюжет?

Нескончаемая война бушевала в бескрайнем космосе… Ой, на сюжет плюньте. Понятно, что при таких-то родителях, *Alien Arena* – типичная стрелялка от первого лица, в которую обычно играют онлайн, с незнакомыми партнерами; «убей, или убьют тебя» – вот и весь сюжет. Режимов здесь несколько, но каждый посвящен полному уничтожению других игроков. В режиме Capture the Flag (Захват флага) необходимо захватить вражеский флаг на чужой базе и доставить его на свою. Вариант Deathball (Смертельный мяч) дает передышку от череды убийств, ради пробивания «мяча смерти» через вратарскую площадку. All OUT Assault – тот же режим Death Match, обожаемый всеми игроками Unreal Tournament и *Quake III Arena* (отсюда и *Arena* в названии игры). Побеждает игрок, на любимом оружии которого оказалось больше всех зарубок. Игра гораздо ближе к *Unreal Tournament (UT)*, чем к *Quake III Arena*, а некоторые уровни, похоже, создавались с оглядкой на *UT*. Имеется, например, древнеегипетский храм, а также множество других пространственных конструкций. Правда, ощущение масштаба здесь не совсем то, что в *UT*, где часто видны и отдаленные холмы. Даже самые крупные уровни в *Alien Arena* уступают привычным в *UT* ландшафтам. Но это вовсе не плохо: получается компактная напряженная игра, и за каждым углом прячется противник – стоит промедлить с прицелом, и вы уже повержены наземь. Отсюда понятно, почему каждый игрок ведет себя так, словно босиком перебегает раскаленные пески; скоро и вы привыкнете, сжимая пространство, совершать прыжки вместо шагов.

По каждой местности, как полагается, разбросано оружие (в некоторых режимах вы сразу получаете весь арсенал). Как и в *UT*, наш фаворит – вездесущая ракетница с

Наш любимый набор



› **ДОСПЕХИ** **Помогут остаться единственным выжившим в рукопашной схватке.**



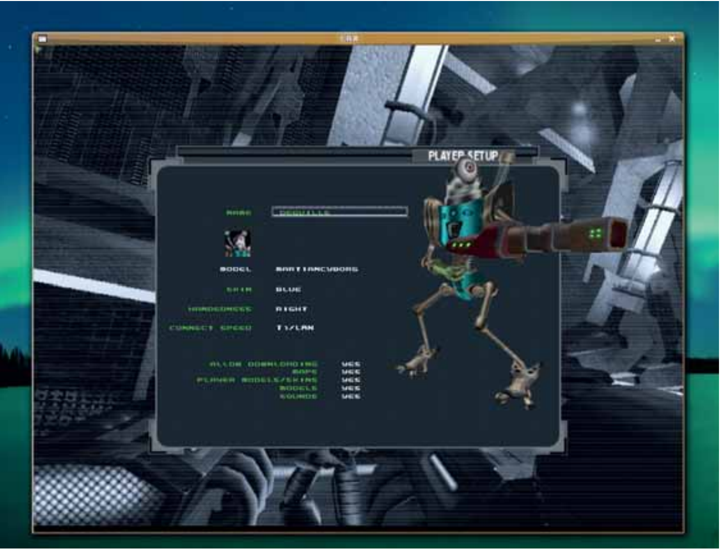
› **РАКЕТНИЦА** **Наше любимое оружие, основа тактики Команды LXF. Цельтесь в пол под ногами противника.**



› **ПУЛЕМЕТ** **В отличие от *UT*, на разминку времени нет: открыть пальбу можно почти сразу. Вражеские полчища выкашиваются подчистую.**



› Превосходное качество игровой графики опровергает мнение об отсутствии приличных художников в мире открытого ПО.



› Ваш персонаж – сущее чучело, но это из-за вторжения врагов. Если выживете, то заработаете и новое лицо, и прочие улучшения.

обширной зоной поражения: она не требует снайперских навыков, достаточно направить ствол в нужную сторону, и в любом случае враг не обойдется без потерь. Есть различные пушки и пулеметы, снайперская винтовка, аннигилятор и минный заградитель. Каждый вид оружия обладает альтернативным режимом, по умолчанию привязанным к правой кнопке. Обычно это гранатомет или снайперский режим.

Еще одна восхитительная вещь – наделение сверхсилой. Налицо привычные супер-мегаразрушительность и неуязвимости разных типов, но есть и падение гравитации, позволяющее перепрыгивать через уровни, и ускорение, придающее вашему персонажу амфетаминовую стремительность. Каждый раунд обычно начинается с отчаянной борьбы за оружие, которое разбирается мгновенно: замешкался – считай, не жилец!

Боты для битья

Если у вас проблемы с выживанием, то есть превосходный способ потренироваться в одиночных режимах с механическими противниками-ботами. Они специфичны для каждого уровня и являются легкой мишенью даже при самых слабых настройках. Конечно, это противники не из трудных, но приобрести игровой опыт в *Alien Arena* иначе не получится: вступив в игру онлайн, вы тут же будете биты, не сообразив, где урвать ракетницу. Многие онлайн-игроки уже имеют некоторую практику, и хотя мы нашли их довольно дружелюбными, способности большинства вызывают трепет. А ведь в сетевом состязании – главная прелесть этой игры, и всегда найдется несколько человек, одновременно играющих в нее онлайн; что и делает ее одной из лучших стрелялок для нашей ОС (по крайней мере, до выхода *Unreal Tournament 3*).

Но лучшее в *Alien Arena* – свобода: как игры, так и совершенствования. Исходный код – интересное чтение для хакера; кроме того, вокруг создания карт и уровней игры уже сложилось сообщество любителей. Каждый может создавать модели, рисовать собственные текстуры и программировать ландшафты, и чем больше людей делают это, тем игра сильнее. Ландшафты, поставляемые вместе с игрой, можно настраивать по своему вкусу, а это возможно лишь тогда, когда игрок имеет доступ к самым основам замысла. Именно этим и берет *Alien Arena*: фантастическая увлекательность, сдобренная свободной разработкой. Если вы ищете стрелялку для свободной операционной системы, то *Alien Arena* – один из лучших вариантов. LXF



› Можно воспользоваться любым средством передвижения, чтобы побольше навредить своим непрошеным гостям.



LINUX FORMAT Вердикт

Alien Arena 2007

Разработчик: COR Entertainment
Сайт: www.alienarena.org
Цена: GPL

Сюжет	8/10
Графика	6/10
Увлекательность	7/10
Документация	7/10

Рейтинг 7/10

CrossOver Professional 6.2

Windows-компьютер ныне можно запустить внутри виртуальной машины из-под Linux, и зачем тогда подобные программы? Размышляет **Энди Ченнел**…

Вкратце...

» Без особой возни эмулирует рабочую среду Windows для определенных приложений и игр. См. также: *Cedega*.



Грэм считает...

«По-моему, разработчикам *CrossOver* не стоит сворачивать работу прямо сейчас, когда проект *Wine*, на котором основан *CrossOver*, для многих пользователей важен, как никогда.»

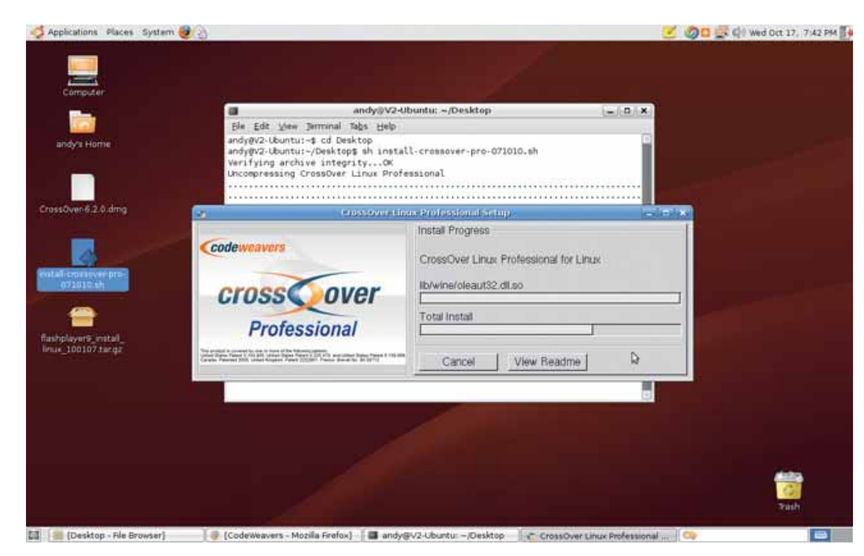
› Интеграция с рабочим столом отменная, а вот поддержка *iTunes* пропала. Досадно, если хочется купить музыку в магазине iTunes.



В нашей редакции *CrossOver* появился с момента создания: нас радовал запуск *Photoshop 7* или *Dreamweaver MX*, мы бешено рукоплескали, когда на рабочий стол Linux пришел iTunes, и облегченно вздыхали, когда Office 2000 наконец-то открывал эти пошлые **.doc**, полученные от других. Каждая версия брала новые рубежи, и Linux-пользователи все реже перезагружались, а раздел Windows все чаще удалялся за ненадобностью. Но время шло, Linux-приложения улучшались, и мы стали замечать, что перестаем пользоваться *CrossOver*.

CrossOver устанавливается по-прежнему быстро и легко: пользовательский интерфейс разделен на две части (для установки и для настройки), доступные из меню Приложения (Applications). Программа может даже добавить ярлыки Windows-программ в меню и запускать их обычным способом. Неофициальное тестирование показывает, что производительность под *CrossOver* программ типа *Dreamweaver*, *Photoshop* или *MS Office* мало уступает таковой в среде Windows XP, а иногда кажется чуточку выше. Ранее функция автосохранения в *MS Office 2000* нередко оставляла Windows-пользователя наедине с пустой строкой, где должен был быть текст. Вновь замолотив по клавишам, бедняга видел старую строку, возникшую впереди набранной. Теперь это в прошлом, экранные проблемы *Wine* устранены, и все работает как надо. Проблематичные приложения могут запускаться в собственной «бутылке» (bottle) – отдельном экземпляре *Wine*, подстроенном к конкретной версии Windows. Работает неплохо, и зачастую программа сама предлагает создать новую «бутылку», предвидя появление проблем: пустяк, а приятно. Любители запускать игры сразу заметят рост производительности. Похоже, разработчики особо довольны успехом *World of Warcraft*: ее скорость превзошла результат *Cedega* (правда, «родная» версия все-таки круче).

Реальная проблема *CrossOver* – узкий диапазон поддерживаемого программного обеспечения, что, впрочем, неизбежно: вопрос этот вне компетенции разработчиков *Wine* и *CrossOver*, которые должны постоянно ловить черную кошку в темной комнате. К примеру, поддержка *Photoshop 7* очень устойчива, а с мелкими проблемами вполне можно ужиться.



› Приложение доступно в виде RPM либо Tar-архива и запросто устанавливается.

О грустном…

Увы, *Photoshop 7* – программа пятилетней давности, и его обновляли уже три раза: мы пробовали *Photoshop CS3*, но он рухнул еще на этапе установки, как и *Dreamweaver*. *Internet Explorer 7* – шаг вперед по сравнению с архаичным *IE6* – тоже споткнулся на первом же шагу. Здесь-то *CrossOver* и уступает ПО виртуализации, запускающему полноценную версию Windows, а значит, имеющему доступ к более широкому набору программ. Одной из задач, ради которых мы в **LXF** применяли *CrossOver*, было тестирование корректности отображения web-страниц в *IE6*. С *IE7* ему уже не справиться – тут настает черед виртуальной машины.

При потребности лишь в тех немногих Windows-программах, которые поддерживает *CrossOver*, цена *CrossOver* Pro на фоне лицензионных отчислений Window XP/Vista и цены пакетов виртуализации вполне способна привлечь. Но не надейтесь насладиться новейшими версиями Windows-систем. Что *CrossOver* берется делать, он делает безупречно – но, памятуя об ограничениях, до покупки загрузите демо-версию и посмотрите, хорошо ли будет вести себя нужная вам программа.

Хоть *CrossOver* и неплохое приложение, а команда Codeweavers – подлинная опора движения открытого ПО, но *VMware* и *Parallels* предлагают ПО виртуализации, которое стоит ненамного дороже (правда, при наличии лицензии Windows), и благодаря прогрессу процессоров и падению цен на память, пользователи не очень-то заметят возросшую требовательность новых версий Windows к ресурсам. LXF



CrossOver Professional 6.2

Разработчик: CodeWeavers
Сайт: www.codeweavers.com
Цена: $39,95

Функциональность	6/10
Производительность	9/10
Простота использования	7/10
Оправданность цены	7/10

» *Отличному проекту грозит вымирание: вот-вот затопчут виртуальные слоны…*

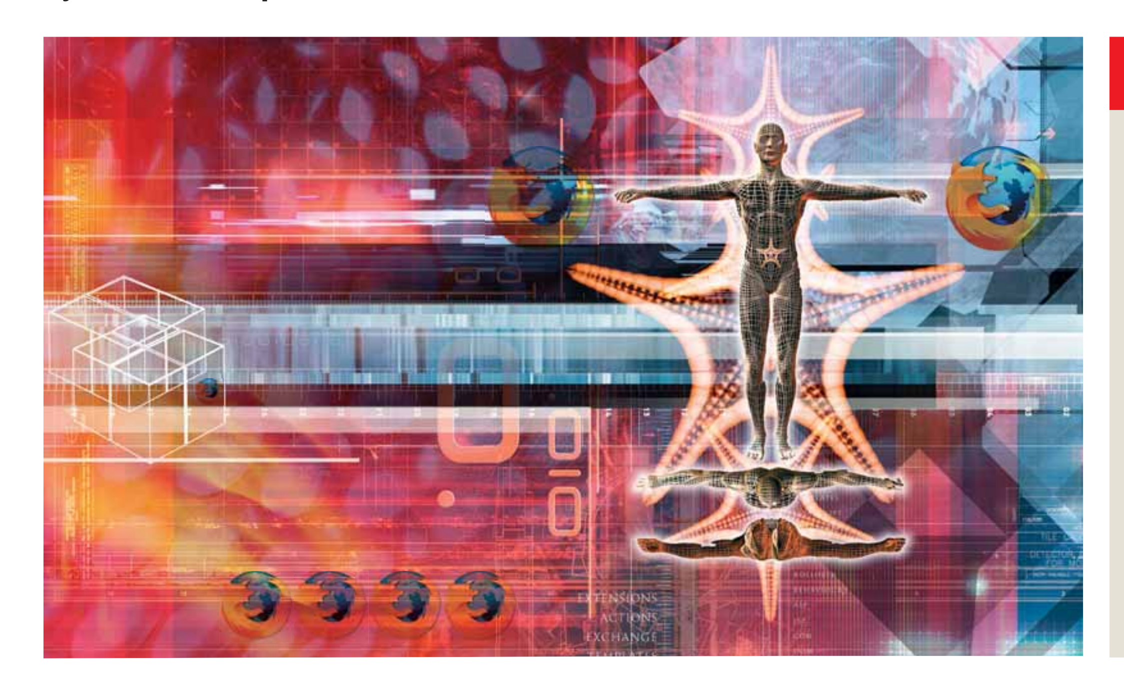
Рейтинг 7/10

Сравнение

Каждый месяц мы сравниваем тысячи программ – а вы можете отдыхать!

Расширения Firefox

А вы знали, что можете добавить расширения в Firefox? Ах, знали… А вы уверены, что выбрали лучшие? **Ричард Коббетт** позаботится об этом.



Про наш тест

Мы успели побаловаться со всеми расширениями, кроме самых распоследних версий. Все они бесплатны и доступны онлайн за один клик. Мы сфокусировались на том, насколько хорошо они выполняют свою конкретную задачу – а если она вам не нужна, то и устанавливать их незачем! Все расширения проверены в работе на простом Ubuntu Live и «голой» копии *Firefox 2* на предмет стабильности, но другие версии также, в основном, поддерживаются, и большинство расширений уживаются друг с другом. Вы также можете столкнуться с более поздними сборками или более ранними версиями *Firefox*. Расширения предназначены для кросс-платформного использования, и годятся также под Mac и Windows.

Наш выбор

В увесистом списке *Linux Format: Миссия невыполнима*, задача «рассказать обо всех расширениях *Firefox*» значится под №45: после «приклеить мармеладного кролика на спину настоящему кролику» и «протащить затасканную метафору мимо недремлющего ока многострадального литредактора **LXF**». Что ж, с учетом последнего предложения, будем считать, что уже под №44.

В данном *Сравнении* мы будем более глобально освещать и выбирать темы из самых разных областей. При всей нашей любви к апплетам-приколам типа Abe Vigoda Death Clock [Когда помрет Эйб Вигода] или примочек к обычным вкладкам, сказать про них особо нечего: вы скачиваете их одним кликом мыши, перезапускаете *Firefox*, они включаются, вы улыбаетесь и возвращаетесь к своим делам. Вместо этого мы придирчиво отобрали пару побочных проектов, у которых недавно вышли крупные релизы, с утилитами, которые пригодятся вам каждый день. Пара из них забавны, большинство – полезны, и все заслуживают внимания как показатель развития *Firefox*.

Другой момент связан с манерой работы *Firefox*: некоторые из более старых расширений могут не работать с новыми версиями. Один из важнейших инструментов, который мы специально не упомянули – *Nightly Tester Tools* (его можно скачать с www.oxymoronical.com/web/firefox/nightly) – позволит установить, происходит это по причине реальной несовместимости версий или просто нужно сменить номер.

Следует помнить, что не существует такой вещи, как окончательное расширение. Ключевая концепция *Firefox* гласит, что никто, от Red Hat до *Opera* и даже до большого



страшного Microsoft, не может предсказать, чего именно вы захотите от вашего браузера. Загребите все подряд в ваш *Firefox* – и вы получите такую раздутую программу, которой и свет не видывал. Всегда старайтесь устанавливать расширения последних версий, причем по одному за раз: так вы сразу заметите, где возникли проблемы, и решите их на месте.

Mouse Gestures

Назад и влево. Назад и влево...

Бойтесь этого расширения... Но в хорошем смысле: скорее как «осторожнее со сладким», чем «осторожно, акулы». С тех пор, как *Opera* применила жесты мышью в web, миллионы пользователей осознали, на свою беду, что мало какие доработки интерфейса вызывают также же привыкание, как открытие страниц и переходы вперед и назад между ними посредством росчерков курсором. Может, и не на уровне голливудской Резолюции меньшинства, но эффект примерно тот же. И если вы пересядете за компьютер, на котором они не установлены, то почувствуете, что угодили в далекое прошлое.

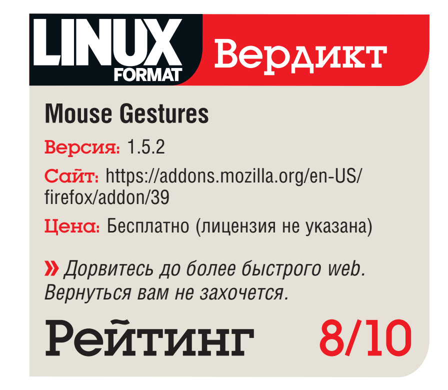
Реализация в *Firefox* более неповоротлива, чем в *Opera*, но вполне приемлема. Когда вы хотите сделать жест, нужно нажать на правую клавишу и шаркнуть мышью одним из условленных способов – например, вправо-вверх и влево открывает все охваченные росчерком ссылки во вкладках. Иные используют колесико мыши для навигации вперед-назад между вкладками, в зависимости от нажатой кнопки.

Изучение всех движений мыши требует времени, но на практике все их знать и не надо. Вы можете запасть на режим, когда мышь оставляет на экране красную траекторию, в пояснение того, что именно вы делаете. Если заданные по умолчанию движения неудобны для вас, можно настроить их самим. Единственно, что если вы захотите вернуться к старым настройкам, то обнаружите: они хранятся в закодированном виде – типа «*RUL» вместо простого «Нарисуйте это...». В общем, не такая уж неразрешимая проблема, но в следующей версии можно сделать и поудобнее.

> «Возврат к работе без них вызывает ощущение, что вы угодили в прошлое.»



› Представьте, что курсор мыши двигается. Жест распознали? Можете сами создать их — с поправкой на свою криворукость.



LINUX FORMAT Вердикт

Mouse Gestures

Версия: 1.5.2

Сайт: https://addons.mozilla.org/en-US/firefox/addon/39

Цена: Бесплатно (лицензия не указана)

» *Дорвитесь до более быстрого web. Вернуться вам не захочется.*

Рейтинг 8/10

Flock

Расширений столько, что назвать это Firefox'ом язык уже не поворачивается.

Это даже не расширение, а революционное преобразование – просто полностью иной *Firefox*. Идея простая: все старые преимущества работы в Интернете совмещаются с совершенно новыми, делая *Firefox* более «гладким», с большим набором функций «прямо из коробки», готовым взаимодействовать web-сервисами. Вместо обычных закладок, он получает информацию прямо с del.icio.us от Yahoo. Вместо боковой панели, предоставлявшей лишь закладки и историю, вы получаете встроенный доступ к таким сайтам, как Twitter, Facebook, MySpace, Flickr и YouTube. И так далее.

Полученный браузер напоминает *Opera*, хотя бы потому, что в отличие от большинства браузеров, все файлы программы умещены в одну закачку. У вас будет все тот же доступ к Темам и Расширениям, и другие функции *Firefox* тоже, но главный повод установить *Flock* – упаковка громких имен Web 2.0 в один тщательно сбалансированный пакет.

И в этом *Flock* преуспел. Элементы, подобные Media Bar – это панель размером с окно, которая заполняется информацией с ваших учетных записей и с таких сайтов, как Flickr – интегрированы поразительно удачно. Встроенная читалка RSS – одна из лучших нами виденных, она позволяет сохранять истории и сортирует ленты. Боковая панель теперь стягивает данные отовсюду, от Twitter (чтобы друзья были в курсе, чем вы занимаетесь) до del.icio.us (для хранения закладок). И если вы ведете блог, возможность надергать кусочков из сети и опубликовать их на вашем сайте сэкономит вам много времени.

Конечно, если вы не в Web 2.0, то вам все это безразлчно. *Flock* очень удобен в использовании, но даже таким заядлым любителям Интернета, как команда **LXF**, просто не нужно столько навороченных функций сразу, а наши любимые расширения для самого *Firefox* будут развиваться и дальше.

> «Конечно, если вы не в Web 2.0, то вам все это безразлично.»



› Дико извиняемся за выражение «Web 2.0». Желающим предложить альтернативное название просьба писать на известный адрес.



LINUX FORMAT Вердикт

Flock

Версия: 1.0

Сайт: www.flock.com

Цена: бесплатно (лицензии MPL/GPL/MGPL)

» *Для интересующихся данной сферой в этом первом крупном релизе есть на что посмотреть.*

Рейтинг 7/10

AdBlock Plus

Если оно у вас не установлено, вы неправильно используете Firefox. Или – вы пожиратель рекламы…

А *dBlock Plus* – лучший способ избавиться от рекламы в браузере. Можно было бы и не продолжать: этим все сказано. Это настолько полезное, распространенное расширение, что оно даже вызвало войнушку против *Firefox*, и такое удобное, что вас скоро начнут приводить в недоумение жалобы на рекламу в Интернете. «Баннеры? Это вы о чем? А-а, всплывающие окна! Ох, простите…»

Как и большинство приложений, подавляющих нудную рекламу, *AdBlock Plus* постоянно воюет с людьми, которым действительно нужно, чтобы мы смотрели все эти объявления о сайтах знакомств и бесплатных обоях для рабочего стола. После установки вы будете автоматически подписаны на один или более списков, регулярно снабжающих вас последними новыми фильтрами, без вашего вмешательства. Они справятся с большинством раздражающих объявлений, с которыми вы столкнетесь. Можно создать и свои фильтры, направленные на конкретную рекламу – или блокируя целые сайты, или по ключевым словам. Если что-то все же просочилось, просто нажмите правую клавишу мыши и выберите AdBlock Image из контекстного меню.

Инструменты типа *AdBlock* очень важны не только для сохранения рассудка, но и для комфорта работы в Интернете. Уместно возразить, что подавляя рекламу, вы лишаете дохода владельцев сайтов – жаль, конечно, но нельзя не признать тот факт, что если бы реклама из Интернета так неистово не бесила людей, они не стали бы ее и убирать. Заметим, что лишь немногие блокируют текстовую рекламу от Google, свидетельством чему – приличный показатель «проклика». Равных ему средств борьбы не существует – каждый раз, когда реклама проходит через фильтр, недолго ждать, пока что-то другое не окажется на ее месте. Если *AdBlock* потерпит неудачу, его наследник будет непобедим. Пусть слоганом *Firefox* остается «Вернем Web» – именно инструменты, подобные *AdBlock*, помогают внедрить его в жизнь.

> «Баннеры? Бан... неры? А что вы называете “баннерами”?»



› Не отображено: масса раздражающей рекламы. Невероятно, правда? Привыкайте к хорошему!



StumbleUpon

Прервитесь. Исследуйте «закоулки» Интернета в Сизифовой погоне за инфорадостями.

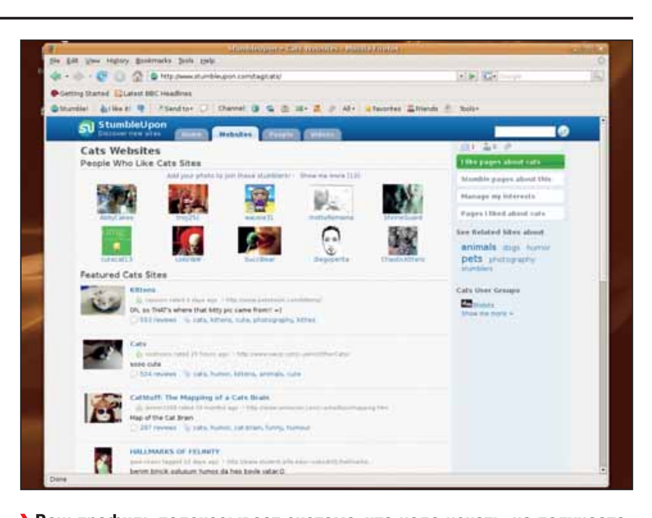
Д аже по самым консервативным оценкам, интернет – Реально Крутая Штука. Независимо от ваших интересов и объемов свободного времени, всегда найдется сайт, способный превратить продуктивный день в сериал из судорожных кликов. И это прекрасно. Другое дело, что большинство людей регулярно посещают только небольшое число сайтов, и найти что-то действительно новое куда сложнее, чем кажется. *StumbleUpon* – это величайшее расширение, предлагающее руку помощи ради шанса разгневать вашего босса/учителя/жену/кошку.

После установки вы увидите новую панель с тремя главными кнопками: Stumble – чтобы открыть что-то наугад, и Thumbs Up/Thumbs Down – чтобы выставить оценку открывшейся странице. Thumbs down поищет вам более интересную замену, Thumbs Up отметит ваше одобрение и добавит вероятности отправить на этот сайт других людей.

Кроме обычного просмотра страниц, есть также некоторые организационные функции, но вы их вряд ли будете часто использовать. Можно оставить отзыв на посещенную вами страницу, создать очередной список единомышленников и очередную коллекцию закладок. Единственное, что действительно полезно – это возможность указать, какие типы сайтов вас интересуют. Категории достаточно широки: от Активизма до Буддизма, и довольно силен технический уклон. Изредка вы получаете шальную страницу по случайному выбору, но это только добавит веселья.

И что же можно получить? Клик. Maininthedark.com – Flash-анимированный резиновый человечек, который тащится за курсором мыши. Клик. Четыре лошади, при нажатии запевающие хором. Клик. Как Отшить Нахала. Клик. Открытое сообщество дизайнеров. Клик. Клик…

> «Что там у нас? Посмотрим. Клик. Клик. Клик. Клик. Куда девался день?!»



› Ваш профиль подсказывает системе, что надо искать, но получаете-то вы выбор «от рулетки».



Greasemonkey

Расширение одно-за-всех; это вам не Abe Vigoda Death Clock.

Среднестатистическое расширение добавляет пару функций к браузеру, а *Greasemonkey* входит в DOM само, исполняя все – от подавления рекламы до выборки информации: например, соединяет Gmail и Google Reader в одну отличную платформу сообщений, и даже встраивает новые функции типа скачивания за один клик видео с Youtube или фильтрации результатов поиска изображений в Google.

Полноценные расширения часто лежат за пределами способностей к программированию многих людей, но *Greasemonkey* полон простеньких сценариев JavaScript. Пусть вы и не замахнетесь на капитальную реформу Google Reader, но почти каждый может набросать скрипт, чтобы, например, навсегда скрыть с глаз надоевшего участника форума или убрать особо навязчивую рекламу. Расширение берет на себя черную работу: каталогизирует установленные скрипты, автоматически добавляет новые, какие вы хотите, и упрощает их запуск и отключение по запросу.

Чем полезно *Greasemonkey* – это экономией на числе установленных расширений, а стало быть, уменьшением шансов на ошибку. Большинство готовых скриптов написаны для определенных сайтов (или web-сервисов), и они не будут бодаться с вашим браузером, а вам незачем брать в голову, как, собственно, эта магия работает. Однако помните об этом, устанавливая новые скрипты – убедитесь, что им можно доверять. Раньше хватало «подлых» скриптов, например, отправлявших cookies в злые руки, а новорожденные скрипты часто конфликтуют с сайтами, вместо того чтобы сотрудничать с ними. Короче, если надо взять только одно расширение на необитаемый остров, то оно перед вами.

> «Если на необитаемый остров надо взять только одно расширение – вот оно.»

› Вам незачем пахать за web-разработчиков, но уж если придется, почему бы не учесть собственные интересы...



LINUX FORMAT **Вердикт**

Greasemonkey

Версия: 0.7.2
Сайт: www.greasespot.net
Цена: Бесплатно (MIT)

» *Засучите рукава и примените свои любимые правила к любимым сайтам.*

Рейтинг 9/10

BlueOrganizer

Когда CD – не диск? Когда это способ нырнуть в бескрайний Интернет.

В каждом web-браузере есть закладки, но *BlueOrganizer* зашел дальше. Теоретически, он способен сообразить, что вы ищете: сайт, изображение, книжку, CD, ресторан или что другое, и должным образом посодействовать. Практически – он еще не доработан, и это сразу заметно.

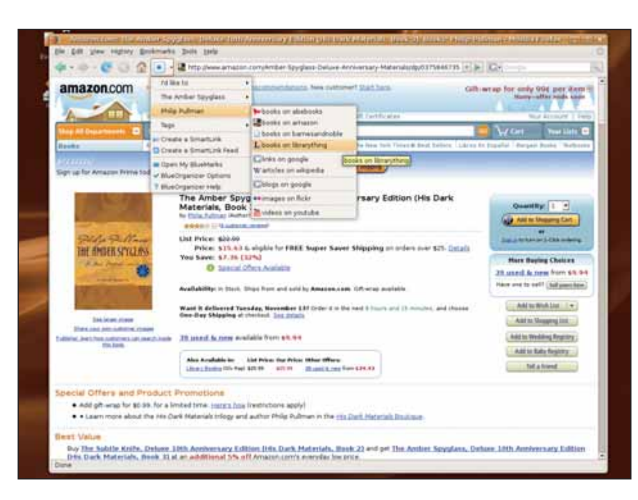
И очень жаль, потому что когда все работает хорошо, это полезное расширение. Оно: **а)** позволяет вручную или автоматически сохранять интересующие вас страницы и размещать в вашем блоге «умные ссылки» SmartLinks, помогающие его читателям переходить на специализированные сайты для получения дополнительной информации по указанному объекту; **б)** пытается распознать на web-страницах медиаобъекты (названия книг, фильмов и песен, имена авторов, актеров и т. д.) и выдает меню со ссылками на эти объекты на тех же спецсайтах. Так, на последнюю книгу Терри Пратчетта, «Делать деньги», вывелось меню со ссылками на:

– LibraryThing – для добавления ее в вашу коллекцию;
– список пожеланий Amazon.com;
– другие инструменты создания закладок;

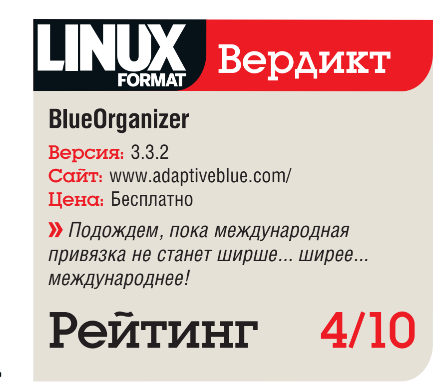
а имя автора и название оказались в собственных меню со ссылками на результаты поиска в Google и статью в Википедии. И добавилось множество тегов, включая «научная фантастика» и «мир CD» для следующих поисков.

Увы, это работает только с Amazon.com. На Amazon.co.uk *BlueOrganizer* книгу не нашел, предложив только общие ссылки на Amazon и заливку ссылки на Twitter. Если возможности расширения позволят работать не только с американскими сайтами, оно станет хорошим дополнением к вашему браузеру. Но пока что нам, бедным сыроядным обезьянкам Третьего мира, от него мало пользы. Можете применить к нему схему «Поживем – увидим».

> «Не будь он заточен под США, стал бы полезным добавлением к браузеру...»



› Хотя он отлично работает с Amazon.com, для Amazon.co.uk гораздо меньше возможностей.



LINUX FORMAT **Вердикт**

BlueOrganizer

Версия: 3.3.2
Сайт: www.adaptiveblue.com/
Цена: Бесплатно

» *Подождем, пока международная привязка не станет шире... ширее... международнее!*

Рейтинг 4/10

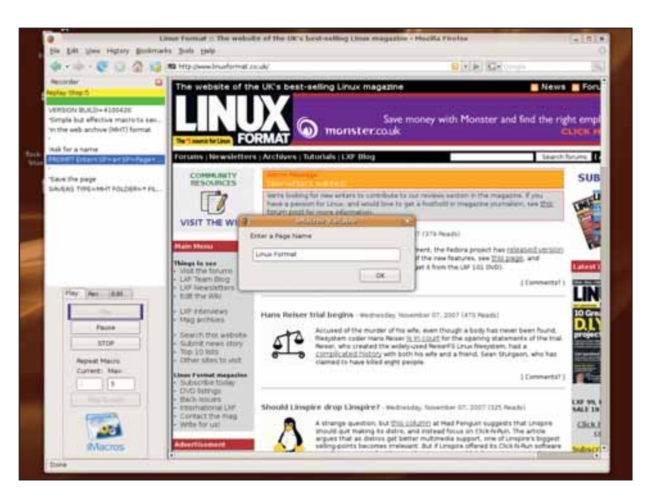
iMacros

Напишите, повторите. Напишите, повторите. Напишите, повторите. Напишите…

Firefox – это не только просмотр web-страниц, но и мощное подспорье для всей вашей жизни. Финансы, коммуникации, сбор информации… он добудет все, что есть в Интернете. Расширение *iMacros* облегчает создание макросов для черновых работ: получения информации с сайтов, заполнения бланков или просто навигации по Интернету. Фактически, данное расширение *Firefox* – рекламный образец коммерческих инструментов web-редактирования от iOpus, но вполне функциональный, так что все OK. Оно бесплатно и для частных пользователей, и для коммерческих целей, а не хватает в нем только функций высшего пилотажа, например, автоматизации Flash-апплетов. Зато есть Вики, набитая советами и примерами макросов. Это очень важно: можете создать новый макрос, просто нажав на запись и просматривая страницы в обычном режиме, но главная мощь *iMacros* проявляется, когда вы решите приложить руки к милому маленькому JavaScript. Простое сохранение и надежда на лучшее могут привести к проблемам, особенно когда в задачу входят несколько закладок и web-страниц.

Результат выглядит необычно: ваш макрос показан слева, а главное окно, похоже, постоянно бегает на сайт iOpus за информацией. На практике все работает на вашем компьютере, а расширение перехватывает запросы. Сейчас еще нет большой коллекции, из которой можно черпать, хотя есть группа del.ici.us, там много чего хорошего. Интересно, что вам не нужно скачивать макрос для его использования – его запустит простое нажатие на ссылку, облегчая создание большого архива, доступного с любой сборкой *Firefox*.

> «Финансы, коммуникации, сбор информации... добудет все, что есть в Интернете.»



› Ну-ка, глянем… бланк для данных… походы на web-страницы… черт! Нет макроса «написать обзор».



iMacros

Версия: 6.0.0.1
Сайт: www.iopus.com/imacros/firefox
Цена: Бесплатно

» *Если вам нужны Закладки, Web-данные, Оповещения, Виджеты или Социальные сети, попробуйте.*

Рейтинг 7/10

Web Developer Toolbar

Швейцарский нож для редактирования web-сайтов и лучшая панель инструментов.

Если вы балуетесь созданием сайтов, Web developer Toolbar – то, что вам нужно. Не можете понять, почему ваши ID путаются друг с другом? Нажмите на кнопку для выделения и разметки любой части страницы. Хотите подглядеть CSS на какой-нибудь web-странице? Еще один щелчок накопает вам все таблицы стилей и поместит их на отдельную страницу. Надо бы взглянуть на вашу страницу при другом разрешении экрана? Опция Resize уменьшит *Firefox*, ожидая, когда вы нажмете Обновить.

Все предлагаемые функции просты, но хорошо продуманы. Самая полезная – инструмент Редактирование CSS, позволяющий экспериментировать со страницей, добавляя новые стили и всячески измываясь над вашим кодом. Когда все готово, можете скопировать изменения в основную таблицу стилей. Если все рухнет у вас на глазах, нажмите Обновить – и вернетесь к первоначальному виду. А так как изменения происходят только внутри *Firefox*, не бойтесь, что вы повредите сайт.

Естественно, это расширение не лишено ограничений. Вам так или иначе придется тестировать изменения в других браузерах: если страница отображается корректно в *Firefox*, это еще не гарантия, что так будет в любом браузере. Но большинство инструментов полезны независимо от платформы, на которой вы работаете, особенно по части деталей. Замена изображений их ALT-атрибутами укажет те, что нуждаются в доработке, а инструмент Линейка позволяет уточнить размеры и подтвердить положение фиксированных элементов.

> «Создает смысл завести Firefox, даже если вы обычно работаете в других браузерах.»



› Идеально для любых web-разработчиков при поиске конфликта классов или забытых тегов <div>.



Web Developer Toolbar

Версия: 1.1.4
Сайт: www.chrispederick.com/work/web-developer/
Цена: Бесплатно (GPL)

» *Полезный инструмент для любого web-разработчика, независимо от сложности разрабатываемых сайтов.*

Рейтинг 9/10

Расширения Firefox

Вердикт

Greasemonkey 9/10

В *Сравнении* вроде нашего победителя как такового назвать нельзя. Рассмотренные расширения не конкуренты друг другу, их можно свободно скачать, и они могут использоваться как вместе, так и по отдельности. Вы можете установить их все, или не устанавливать ни одного, или установить какие-то другие.

Так почему же мы решили выбрать *Greasemonkey*? А просто это прекрасное дополнение к *Firefox*. Браузеры – они как полотно, а не просто инструменты; они развиваются и становятся мощнее с каждым новым расширением. Они – оружие для изменения баланса сил. *Firefox* создавался для атаки на старый Web, где правил *Microsoft Internet Explorer*, со всей его застойностью. Много лет Microsoft даже не задумывался об обновлениях своего браузера, даже если он трещал по швам от уязвимостей, не предоставлял нужных функций и не пытался расширить базовую структуру Web в каких-то аспектах, где мы до сих пор стараемся разработать что-то новое. И хотя *Firefox* пока еще младший игрок, его влияние очевидно. *Internet Explorer 7* – а это на удивление хороший браузер, если вспомнить недочеты предыдущих версий – можно назвать таким же детищем Mozilla, как и тяжкого труда его собственных разработчиков. Расширения браузера, которые на протяжении многих лет были вотчиной spyware, бесполезных поисковых панелей и прочих «кружев» корпоративного бренда, снова стали чем-то впечатляющим. Баланс сил сместился. Вместо использования браузера, навязанного разработчиками, мы получили то, что хотели увидеть.

> «Firefox создавался для атаки на старый Web, где правил Microsoft Internet Explorer, со всей его застойностью.»

Greasemonkey – это следующий уровень. Это переход от того Интернета, который создали разработчики, к тому Интернету, какой мы считаем правильным. Нужна дополнительная функция? Добавьте. Что-то не нравится? Долой. *Greasemonkey* и его предшественники вроде *AdBlock Plus* добились качественного скачка в Web – от статической, неизменной коллекции страниц к мировому хранилищу информации, где у пользователя больше прав, чем у разработчика. Это заметно во всем: в RSS и в стартовых страницах, web-поисковиках и их движках, в браузерах, подобных *Opera Mini*, которые переводят целые страницы на новые платформы, для которых изначально не были задуманы.



› Скрипты *Greasemonkey* для некоторых сайтов не панацея на долгое время: сайты обновляются постоянно.

Подобные изменения всегда происходят медленно: против них будут неистово бороться контент-провайдеры, рекламщики и все те, кто имеет выгоду от текущей ситуации. Как бы то ни было, если вы уже успели понять преимущества возможности порулить, вы не скажете, что за этим нет будущего. А с расширениями вроде *Greasemonkey* нет повода не наслаждаться им прямо сейчас. LXF

Ваше мнение?

Какие расширения Firefox вы считаете самыми нужными? Какие не стоят внимания? Предпочитаете ли вы другой браузер *Firefox*? Вы уже попробовали *Firefox 3 beta*? Напишите нам ваш список и рекомендации по расширениям (большим или маленьким), которые, на ваш взгляд, пригодятся другим, по адресу letters@linuxformat.ru, или поделитесь с другими читателями на форуме www.linuxforum.ru.

Характеристики расширений

	Частота обновлений	Размер (КБ)	Лицензия/Цена	Версия	Тип
AdBlock Plus	Регулярно (подписки)	228	MPL	0.7.5.3	Расширение
BlueOrganizer	Нерегулярно	282	Бесплатно	3.3.2	Панель управления
Flock	Активно разрабатывается	10.7МБ	MPL/GPL/LGPL	1	Браузер
iMacros	Регулярно	183	Бесплатно	6.0.0.1	Скрипты
Greasemonkey	Частые релизы скриптов	99	MIT	0.7.2	Скрипты
Mouse Gestures	Заморожено	172	Бесплатно	1.5.2	Расширение
StumbleUpon	Время от времени	212	Бесплатно	3.15	Панель управления
Web Developer Toolbar	Нерегулярно	174	GPL	1.1.4	Панель управления

Distrowatch

Новости этого месяца со всех четырех частей мира разработки



ЛАДИСЛАВ БОДНАР
основатель, редактор, начальник и сотрудник DistroWatch.com.

Novell'у — бойкот?

Подписание Novell'ом бесславного соглашения с Microsoft о защите патентных прав раскололо мир открытого ПО на два лагеря. Первый – и более шумный – немедля призвал к бойкоту продукции Novell, в знак протеста против этой сделки. Однако вторая группа решительно отказалась: после всего, что Novell сделал для сообщества Linux, бойкот OpenSUSE выглядит не очень-то красиво. Кто же прав?

По моему мнению – которое разделяют многие – Novell не прав, подписав соглашение с Microsoft. Сообщество разработчиков Open Source отчаянно борется со вводом патентов на ПО в Европе и других регионах, а Novell (и, как следствие, Xandros, Linspire и Turbolinux) фактически узаконивает претензии Microsoft, гласящие, что Linux нарушил ряд их патентов. Хотя при отсутствии тому свидетельств все это не более чем обычный Microsoft'овский FUD [аббревиатура от Fear, Uncertainty and Doubt – страх, неуверенность и сомнения, – *прим. пер.*].

Итак, надо ли доходить до бойкота компаний Linux, подписавших пресловутое соглашение? Трудный вопрос. Лично я не сторонник широкомасштабных бойкотов целых компаний и их продукции. Но, при нашем-то выборе Свободных ОС и программ, должен признать, что я сознательно предпочел бы использовать дистрибутив от одной из тех компаний, которые открыто и недвусмысленно отвергают патентные претензии Microsoft.

Red Hat, Canonical и Mandriva публично отказались подписывать подобное соглашение, рискуя подвергнуться потенциально разорительным судебным процессам – но своим отпором могучему шантажисту они заслужили уважение сообщества Linux.

ladislav.bodnar@futurenet.co.uk

Пора обновить

Большая релизная гонка четвертого квартала 2007 года завершилась. С каким дистрибутивом Linux вы встретите Новый год?

У разработчиков всех основных дистрибутивов Linux уже вошло в привычку в это время года выпускать новые версии своих флагманских продуктов. Первым в марафон вступил OpenSUSE 10.3 с новой программой установки на одном CD, за ним последовала Mandriva 2008, переработавшая подсистему распознавания оборудования и упростившая инсталлятор. Затем Ubuntu преподнес плавную интеграцию 3D-рабочего стола благодаря *CompizFusion*, а там и Fedora явила свое самое свежее и лучшее, с превосходным оформлением и новым звуковым сервером.

По-вашему, это не столь уж потрясающие функции? Ну, в общем, да. Но это не значит, что они плохи, и тем более не стоит ругать разработчиков. Изначально считалось, что многие из них сконцентрируются на вводе в свои дистрибутивы существенно улучшенного *X.Org 7.3*, а долгожданный KDE 4.0 станет вишенкой на торте. Увы, один оказался до изумления переполнен ошибками, а другой на несколько месяцев задержался. И, не теряя время зря, разработчики переключились на отделку уже существующих стабильных рабочих столов. Результат – гармонизация различных компонентов и более гладкая работа.

OpenSUSE 10.3 стал первым из основных релизов. Помимо обычных изменений в интерфейсе для управления пакетами и нескольких новых модулей *YaST*, он принял концепцию поставки на одном CD и разработал Live CD с возможностью установки. Однако после его выхода множество пользователей и обозревателей сообщили о непривычно большом количестве ошибок, ускользнувших от внимания разработчиков – большая неожиданность, учитывая традиционно высокие стандарты контроля качества, характерные для этого дистрибутива. С другой стороны, этот последний релиз OpenSUSE зашел дальше всех прочих дистрибутивов в интеграции элементов KDE 4.0.

Следующим был Mandriva 2008. И вновь, релиз не произвел впечатления серьезного сдвига; но – возможно, в силу внутренних перестановок персонала (новый директор по инжинирингу, новый ведущий разработчик, новый специалист по ошибкам) – он стал первым за несколько лет релизом Mandriva, заслужившим дружное одобрение со стороны своих пользователей. Это приятный поворот событий для компании, старающейся удержаться на плаву. Кажется, вернулся добрый старый Mandrake со своим ориентированным на сообщество подходом и повышенным вниманием к рабочему столу – вернулся, чтобы занять достойное место среди лучших дистрибутивов Linux на рынке!



› **Столько дистрибутивов… Который же лучше?**

За ними последовал долгожданный Ubuntu 7.10, столь нетерпеливо предвкушаемый сообществом настольной Linux. Почему? А по причине выполненной с хорошим вкусом и ненавязчивой интеграции 3D-технологий рабочего стола, а также мелких улучшений, облегчающих жизнь начинающим – таких, как помощь в переходе с Windows и настольный поиск. И хотя некоторые пользователи сообщили о проблемах с обновлением дистрибутива, большинство обозревателей считают 'Gutsy' одним из самых тщательно отделанных и отлично интегрированных дистрибутивов Linux. К сожалению, те, кто ждал, что родственные проекты Ubuntu – Kubuntu и Xubuntu – встанут вровень со старшим братом, испытали горькое разочарование, обнаружив, что многие из отличных новых функций были разработаны только для Ubuntu и Gnome.

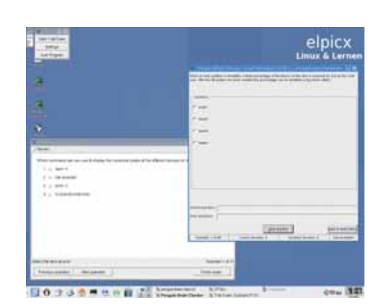
Наконец, настала очередь и Fedora предъявить плоды своих трудов. Как и в остальных трех дистрибутивах, разработчики Fedora 8 решили уделить внимание менее броским задачам – например, улучшению поддержки популярных ноутбуков и интеграции звукового сервера PulseAudio; а команда художников продолжала совершенствовать внешний вид рабочего стола. Fedora по-прежнему остается единственным из основных дистрибутивов, который активно препятствует добавлению небесплатных программ к своему продукту. Сторонники чистоты Free Software от этого в восторге, но любителям всяких модных примочек придется наведаться в репозиторий **Livna.org**.

Стать Linux-гуру

elpicx 1.1 – Live-дистрибутив, который готовит к экзамену на сертификат LPI.

Без сертификата Linux, многим претендентам на ИТ-должность придется побороться за работу своей мечты. Но какого именно сертификата? Есть чудный способ убедиться, что бессонные ночи, проведенные за чтением толстенных мануалов, не пропадут впустую при оценке результатов. Сертификат не зависящего от дистрибутивов Linux Professional Institute (LPI) весьма популярен и труднодостижим: по статистике, экзамены в LPI заваливаются чаще, чем другие экзамены на сертификат, включая Red Hat Certified Engineer (RHEL).

Чтобы помочь вам подготовиться к экзамену LPI, немецкий проект elernit разработал отличный Live DVD, который постарается помочь вам не увеличить статистику провалов. Названный elpicx, загрузочный DVD основывается на Knoppix и CentOS и является идеальным тренажером работы с двумя самыми популярными форматами управления пакетами – DEB и RPM. Обширная документация на DVD включает Linux Fundamentals, экзаменационные билеты LPI, обучающие материалы, руководство по подготовке к экзамену и упражнения: это самая полная подборка обучающих материалов



› **Elpix live DVD с двумя запущенными LPI-симуляторами.**

по Linux, когда-либо опубликованная на загружаемом носителе.

Для проверки ваших новообретенных знаний на DVD имеются два имитатора экзамена: LPISim и Penguin Brain Checker. Оба содержат целый ряд вопросов, подобных тем, на которые придется ответить экзаменуемому во время настоящего испытания в LPI. Количество вопросов и время на ответ можно установить самому, но имеется также опция, весьма точно воссоздающая обстановку реального экзамена. Фантастическая возможность подготовиться к итоговому тесту – и войти в аудиторию уверенным в успехе.
www.elearnit.de/live_cds/elpicx/

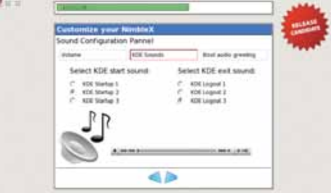
Кликни и прожги

NimbleX – создание личного Live-дистрибутива из дружелюбного GUI.

Подумываете о персонально настроенном Live CD, но лень изучать язык скриптов? Не переживайте. Благодаря проекту NimbleX создание Live CD стало проще, чем когда-либо; если вы умеете пользоваться мышью, то сможете скомпилировать собственный Live CD, включив в него любые драйвера устройств, пакеты ПО, функции и локализацию языка по своему вкусу.

Такое стало возможным благодаря web-приложению, написанному на PHP Богданом Радулеску [Bogdan Radulescu]. Оно проведет пользователя через ряд шагов, включающих выбор пакетов и различных опций настройки, а затем укажет кнопку для создания ISO-образа. Через несколько минут Live CD будет готов для загрузки. Это настоящий Live CD на базе Slackware, но в отличие от типового образа, созданного разработчиком, он настроен под потребности конкретного пользователя.

Самая впечатляющая функция проекта – Custom NimbleX, но ею дело не заканчивается. Специальное издание NimbleX под названием sub_100 является первым Live CD, содержащим рабочий стол KDE и при этом занимающим менее 100 МБ. Ну, а если кому-то просто нужен обычный Live CD на базе Slackware, всегда есть стандартное издание NimbleX – CD размером 200 МБ, содержащий KDE и *Fluxbox*, отличную поддержку беспроводных сетевых соединений, полную поддержку записи NTFS и прочие полезные функции.
www.nimblex.net





› **Настройте различные параметры рабочего стола, включая звуки KDE и фоновые обои.**

Статистика использования дистрибутивов

Какова доля на рынке различных дистрибутивов Linux? Хотя стопроцентно точный ответ тут маловероятен, все же существуют данные, способные пролить свет на проблему. Это, например, данные анализа журналов web-серверов на Linux-сайтах. Поскольку большинство дистрибутивов (за примечательным исключением Slackware и Gentoo) идентифицируются в своих браузерах (модифицируя поле User Agent), то журналы web-сервера позволяют определить их персональные доли.

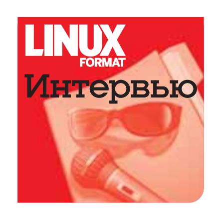
Таблица справа отражает статистику, собранную **DistroWatch.com** в октябре 2007 г. За этот месяц на сайте побывало более 3,7 миллиона посетителей, из них около 40% использовали для доступа на сайт или Linux, или другую ОС Unix. Помимо статистики дистрибутивов, интересно также отметить, что пользователи *Firefox* составляли 58,6%, пользователи *Internet Explorer* – 18,8%, а пользователи Opera – 8,0% от общего количества.

Rank	Distribution	Market share
1	Ubuntu	31.30%
2	Debian	9.90%
3	OpenSUSE	7.40%
4	PCLinuxOS	6.10%
5	Fedora	3.80%
6	Mandriva	3.50%
7	Linux Mint	3.40%
8	MEPIS	1.40%
9	FreeBSD	0.80%
10	Kubuntu	0.70%
11	CentOS	0.40%
12	Zenwalk	0.40%
13	Red Hat	0.30%
14	Pardus	0.20%
15	Solaris	0.20%
16	Elive	0.20%

Великий диктатор

Jason E Kaplan

«Великодушный пожизненный диктатор» сообщества Python и разработчик Google **Гвидо ван Россум**, продолжая надзирать за процессом разработки Python 3000, милостиво назначил аудиенцию **Грэму Моррисону**…



Голландский программист, работающий в Google с 2005 года, Гвидо ван Россум [Guido van Rossum] помимо всеобщего восхищения, в 2001 году получил Премию FSF за продвижение свободного ПО, а также был признан Выдающимся инженером Ассоциацией Вычислительной Техники в 2006.

Недавно на OScon Linux Format прервал его ланч, чтобы расспросить о проекте Python 3000, который сосредоточился на удалении дублированных модулей и конструкций из языка Python, чтобы реализовать подход «Один и только один очевидный способ сделать это», нарушая, однако, обратную совместимость с Python 2.x. Окончательный релиз запланирован на август 2008 года.

Linux Format: Как вы попали на работу в Google?

Гвидо ван Россум: Google долго пытался меня заманить. Первое приглашение последовало на OScon четыре года назад; я тогда как раз пообещал перебраться в Калифорнию и начать работу в небольшой только что учрежденной фирме, так что время они выбрали неудачно! А потом к ним начали присоединяться разные люди из сообщества Python, и все они мне рассказывали, какое это прекрасное место работы.

LXF: Потому что Google активно использует Python?

ГвР: Здесь здорово работать и без всякого Python'а! Многие из тех, кого я знал по Python, выполняют немалое количество разработок на Java.

LXF: А Google оплачивает вашу работу над Python?

ГвР: Да. Половину своего времени я трачу на работу с сообществом Python – например, эта конференция, разработка Python 3000 и т.д. А вторую половину я трачу на использование Python для разработки внутренних приложений Google. Это большое удовольствие. В Google очень богатый выбор работы – здесь масса интересных задач.

LXF: Изменился ли ваш подход к языку в целом — и к Python 3000 в частности — из-за того, что вы использовали этот язык в такой среде?

ГвР: Не совсем. Меня даже можно обвинить в том, что я слишком прислушиваюсь к внутренним жалобам в Google насчет реализации Python, когда проблема связана с тем, как Google использует Python, или как его устанавливает, или как применяет этот язык. По большому счету, Python не используется в крупных, публичных проектах Google, предназначенных для всего мира. Его использует подсайт code.google.com. Но Google очень активно применяет Python для исследований и управления системой.

LXF: Почему Google выбрал Python? Может, какой-то ведущий разработчик испытывал к нему особую страсть? Или…

ГвР: Не знаю! Возможно, этот выбор сделал кто-то из самых первых работников. Определенно имеется стандартизация Python – если вам надо написать скрипт, вы пишете его на Python. Сейчас много используется и Perl, и Ruby. Конечно же – тысячи скриптов оболочки, но множество крупных внутренних систем реализованы на Python.

LXF: И как же произойдет переход на Python 3000, если Google так активно использует Python?

ГвР: Медленно… Как Google его использует, это не главное! Крупные предприятия, как правило, любят использовать новые технологии не раньше, чем эти технологии хоть чуть-чуть созреют!

LXF: В своем выступлении вы говорили, что не обязательно начинать с чистого листа при переходе с одной версии на другую. Причина тому кроется в хорошем проектном решении или в том, что Python — язык особый?

ГвР: Трудно сказать… Я думаю, во многом это связано с самой природой сообщества. Perl продемонстрировал, что в его сообществе имеется солидная поддержка полного редизайна; причиной тому послужили долгие годы опыта работы с Perl и тот вывод, что кое-что явно нуждается в пересмотре.

Python на самом деле намного меньше, чем Perl – здесь намного больше библиотек и намного меньше языка; а P3000 скорее фокусируется на языке, чем на библиотеках. Очевидно, будет стандартная подчистка, новая библиотека ввода/вывода, но исключительно в рамках языка – и весьма интенсивно.

LXF: А в чем будет основное различие?

ГвР: Если меня никто не убедит внести еще какую-нибудь радикально новую функцию, я полагаю, что самым большим отличием станут строковые переменные Unicode.

LXF: Почему поддержка Unicode заняла так много времени?

ГвР: Вообще-то Unicode у нас есть; его ввели в 1999-м – а первый релиз с поддержкой вышел в 2000 году. В то время у нас были связаны руки: большинству пользователей Unicode был не нужен, и им не хотелось никаких перемен. Чтобы добавить Unicode в качестве отдельного типа данных, нам пришлось поддерживать конверсии и строковые объекты. Мы также вставили несколько конверсий по умолчанию; например, если у вас есть 8-битная строковая переменная, и вы приписываете ее к строковой переменной Unicode, то, если не выполнены определенные условия, в результате получится ошибка.

ГВИДО О PYTHON 3000:
«В нем несколько очень мощных концепций, скомбинированных вполне понятным образом.»

LXF: Означает ли акцент на Unicode, что сообщество изменилось?

ГвР: У нас очень прибавилось пользователей и членов сообщества в таких местах, как азиатские страны, для которых Unicode куда важнее. В Европе многие пользователи просто обходятся локалью Latin-1, они могут считать, что символы и байты – это одно и то же, и благодаря волшебству, совершаемому переменными среды в большинстве случаев, при печати этих объектов происходит именно то, что и должно происходить.

Нажим стал больше – всемирная паутина изменилась, потому что у web-сайтов появилось намного больше причин поддерживать разные языки: сама идея локали – скорее белый слон. Концепция локали появилась в стандарте С в 1989 году, это глобальная настройка, которая гласит: «Все, что я делаю со своими символами, я делаю в соответствии с определенным набором правил для кодирования», и она влияет на то, в каком формате обозначаются деньги, числа, символы и т.д. Она влияет на значение верхнего и нижнего регистра и то, как вы интерпретируете нижнюю половину кодовой таблицы 8-битных не-ASCII символов – то ли это кириллица, то ли буквы с диакритическими знаками и т.п.

Поскольку Python построен поверх С, у нас такая же идея установки локали, чтобы все потоки и код вашего приложения наследовали эту самую настройку. Вы можете изменить ее, но это относительно дорогая опция. API языковой настройки для С подразумевали, что во время инициализации надо определить свою локаль и оставить ее как есть, а пользователи смогут влиять на параметры через переменные среды.

LXF: Это уж хакерство!

ГвР: Да. Пользователю Интернет такая модель ни к чему – вы не можете ожидать, что все ваши пользователи используют одни и те же кодовые таблицы, если ваш сайт не сугубо местный.

Некоторые каркасы Python обладают мощной поддержкой интернационализации, активно использующей Unicode, особенно вышедшие за последние шесть лет: взаимопонимание улучшилось. Раньше подавляющее большинство совершенно не понимали кодирования символов и путали байты и кодовые точки (code point), когда речь шла о символах… В чем разница? Народ медленно, но верно обучился пользоваться существующими возможностями Unicode…

Емкость запоминающих устройств чудовищно возросла, и изначальный подход «если мы перейдем на Unicode, наши программы будут потреблять вдвое больше памяти» уже не так актуален, как раньше. Многие платформы усовершенствовали поддержку Unicode, облегчив нам жизнь. Web-браузеры в наши дни очень неплохо поддерживают разные языки и Unicode. Но и приложения на одной платформе тоже совершенствуются с каждым новым релизом.

LXF: Будет ли существующая Unicode-реализация совместима с новым подходом?

ГвР: Она совместима. В числе прочего в 2000 году мы сделали поведение объектов Unicode – насколько это было возможно – логически связанным с поведением 8-битных строковых переменных. Строковые объекты обоих видов обладают большим количеством методов для производства с ними различных действий: поиска, подсчета, конверсии, связывания слов. Все эти методы определяются идентично для 8-битных строк и для Unicode-строк, и если вы сейчас пишете программу, содержащую текстовые или буквенные данные, и производите обработку текста, даже сегодня один и тот же код будет работать с Unicode. Неопределенность возникает, если вы начинаете конвертировать Unicode, скажем, в UTF-8 или UCS-2, и записывать в файл. Я рассчитываю, что нам удастся получить в основном совместимые API для файловых объектов при обработке текста.

LXF: Именно об этом вы говорили раньше.

ГвР: Я рассчитываю на простой случай, когда вы открываете файл, просто задав имя файла (что по умолчанию создает вам текстовый файл), и затем проход по строкам этого файла будет вести себя точно так же, как ведет себя сейчас: программа будет видеть, что каждая строка – это строковая переменная Unicode, и файл может быть кодирован в UTF-8, что покамест не поддерживается, однако пользователь совершенно не заметит разницы.

LXF: Над чем лично вы сейчас работаете?

ГвР: Планирую написать стандартную библиотеку ввода-вывода, используя байты и объекты Unicode, по минимуму используя код на С, и проделать большую часть работы касательно буферизации и кодирования, которую я упоминал в своем выступлении о Python. Но при этом сделать ее скорее библиотечным модулем, а не стандартным способом открытия файлов; затем я смогу все это смонтировать, чтобы в некий момент встроенная функция Open вернула объект совершенно другого вида; а потом наступит очередь тестирования и выявления неполадок; надеюсь, большинство проблем решится достаточно быстро.

LXF: Каким образом вы делите задачи? Вы отводите определенную часть сообществу или они сами решают, над чем им интересно работать?

ГвР: Когда мы делали 2.5, у нас была основная группа разработчиков, которые распределяли работу между собой. Это итеративный процесс: мы определяем сферу работы, вносим изменения, проводим ряд тестов для регрессионного анализа… Потом снова намылить, смыть – пока все не заработает, как надо, тогда можно двигаться дальше.

LXF: Как продвигается работа над интерпретатором?

ГвР: Это большой проект – надеюсь, он будет завершен где-то через полгода… Мы уже почти год над ним работаем, и я оптимистично считаю, что мы движемся в верном направлении. LXF



Полный текст интервью помещен на www.linuxformat.co.uk/mag/van-rossum.html.





Вектор в русскую сторону

Русскоговорящее сообщество постепенно набирает вес в окружающем нас мире. Свободный векторный графический редактор *sK1*, разрабатываемый **Игорем Новиковым** и **Вальком Филипповым**, закрепил положительную тенденцию на конкурсе Les Trophées du libre 2007 – впервые за пятилетнюю историю этого мероприятия в списке финалистов оказался проект из экс-СССР. **LXF** не замедлил связаться с лауреатами, чтобы выяснить все подробности.

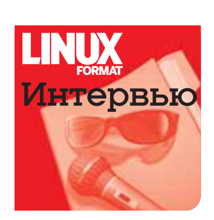
Наши гости



› **Игорь Новиков с наградой («рука с копилефтом») от Les Trophées du libre-2007.**



› **Валёк Филиппов.**



Евгений М. Балдин (LXF): **Представьтесь, пожалуйста.**

ИН: Игорь Новиков (ИН): Род занятий – software engineer или, по-нашему, инженер-программист. Харьков, Украина. Степень магистра по специальности «Биохимия», специализация – генная инженерия.

Валёк Филиппов (ВФ): Системный инженер со специализацией в сетевой безопасности. Степень магистра по специальности «Молекулярная биология и биотехнология», CCIE R&S, CISSP.

ИН: Как видите, команда состоит исключительно из биологов [*улыбается*].

LXF: **Почему именно *sK1*?**

ИН: Имя *sK1* – производное от *Sketch* [свободный векторный графический редактор, позднее переименованный в *Skencil*, – *прим. авт.*]. Заглавная «K» подчеркивает ориентацию на KDE, но библиотечные зависимости от *Qt*/KDE отсутствуют, равно как и от *Gtk*/GNOME. Просто общая компоновка приложения более схожа с KDE'шной, плюс наиболее приемлемыми, на наш взгляд, являются диалоги открытия и сохранения файла, принятые в KDE. Правда, *sK1* может использовать диалоги от KDE, GNOME или Tk, в зависимости от платформы. Стиль виджетов также настраиваемый (он базируется на PNG-шаблонах). Цветовая схема импортируется из текущего менеджера рабочего стола, но может использоваться и своя собственная. Наша цель – создать приложение, которое будет иметь родной (ну, практически родной) вид на любой платформе: GNU/Linux, MacOS X, Microsoft Windows.

Единичка в названии – это то, к чему мы стремимся, т.е. версия 1.0. Неудобно ведь называться *sK0* (улыбается). Если проект выживет, то при смене мажорной версии появится *sK2*, ну и формат файлов поменяется с ***.sk1** на ***.sk2**.

LXF: **Чем *sK1* лучше других векторных графических редакторов?**

ИН: Пока рановато делать громкие заявления. По возможностям импорта мы однозначно лучше многих свободных проектов, связанных с векторной графикой. По производительности и требованиям к системным ресурсам, *sK1* также превосходит все свободные аналоги

(*Inkscape*, *Karbon 14*, *OO Draw*), приближаясь к коммерческим приложениям типа *CorelDRAW* и *Xara*.

LXF: Расскажите об истории создания *sK1*.

ИН: Если вкратце, то дело было так. В 2003 году в Украине начались жёсткие проверки лицензионности программного обеспечения. Продавцы были не готовы к внезапному возникновению спроса на дорогостоящее ПО Adobe, Corel, Autocad и др. «С полки» в лучшем случае можно было купить Microsoft Windows. У многих небольших компаний был выбор между закрытием и попыткой миграции на свободное ПО. Рынок пиратских программ проверки и репрессии не затрагивали, поэтому «пираты» оперативно отреагировали предложением разнообразных дистрибутивов GNU/Linux.

В то время я был совладельцем небольшой полиграфической фирмы в Харькове. Надо было как-то выходить из созданной ситуации. Бухгалтерскую машину перевели на Mandrake 9.1 достаточно легко, т.к. на ней готовилась только первичная документация. А вот с дизайнерскими компьютерами было намного сложнее. В качестве основного инструмента мы использовали *CorelDRAW*. Попытка использовать *CorelDRAW 9.0 for Linux* провалилась с треском: продукт оказался неприемлемо сырым, даже на RedHat 7.0, который был заявлен в качестве поддерживаемой платформы. Поэтому пришлось обратить внимание на свободные векторные редакторы. Выбор пал на *Sketch*, который хоть и не был пригоден для полиграфии, но зато мог импортировать файлы в форматах CMX (*Corel*) и AI (*Adobe Illustrator*). Практически сразу же мы столкнулись с различными проблемами: от несовершенства интерфейса до отсутствия необходимых функций. Чтобы исправлять и дополнять исходный код, пришлось взяться за изучение языка Python, который использовался для написания *Sketch*.

Проект первоначально был предназначен для внутреннего использования: мы не собирались «спасать человечество от засилья Microsoft» или делать что-то там ещё в этом духе. Но Free Software быстро «научило нехорошему», и в начале 2004-го года проект был выложен на **SourceForge.net**.

Впоследствии проблемы в работе фирмы, связанные с предвыборной ситуацией на Украине, вынудили продать бизнес… Я решил стать профессиональным разработчиком и перешёл на работу в американскую аутсорсинговую компанию. Пертурбации в жизни страны («Оранжевая революция») отразились и на проекте. Работа была практически остановлена. Проект снова ожил лишь в конце 2005 года. В команде на тот момент остался только я, как основной разработчик. Остальные участники, по разным обстоятельствам, не смогли продолжать работу над проектом.

В начале 2006-го года активное параллельное участие в проекте новостного портала **LinuxGraphics.ru** (проект Александра Прокудина) привело к выходу первой ознакомительной версии *sK1*. Это произошло благодаря тому, что Александр активно работает в проектах *Scribus* и *Inkscape*. Он сыграл роль инициатора в выпуске ознакомительной версии. Кроме того, он познакомил меня с участниками проекта *Scribus*, а также с Вальком.

Сообщество приняло *sK1* благосклонно, что стимулировало дальнейшее развитие проекта: были реализованы уникальные свойства *sK1*, которые оправдывают его позиционирование как ориентированного на полиграфию. Это, прежде всего, поддержка CMYK-цветов, управление цветом, поддержка CMYK при генерации PostScript. В дальнейшем был добавлен дополнительный рендерер графики на базе библиотеки *Cairo*, что позволило получить субпиксельное сглаживание и поддержку прозрачности.

Примечательно, что в векторном редакторе *Inkscape*, который мы также рассматривали в качестве возможной замены *CorelDRAW*, поддержка CMYK и управление цветами появились в SVN только недавно и будут доступны только в версии 0.46. Переход *Inkscape* на *Cairo* начат и, скорее всего, будет осуществлён полностью к версии 0.47, а в 0.46 *Cairo* будет использоваться только для каркасного режима.

В 2007 году в *sK1* появилась уникальная функциональность: импорт файлов *CorelDRAW* (CDR). Инициатором этого, как и в случае с выпуском первой ознакомительной версии *sK1*, выступил Александр Прокудин. Вокруг этого формата существовало, да и сейчас, судя по форумам, существует масса баек вроде «использовать эти файлы без кода *CorelDRAW* невозможно», мол, это «вещь в себе», что-то из серии «непознаваемого», и т.д. Поэтому изначально был определённый психологический барьер в этом плане. Но, как только мы взялись за работу над форматом, очень быстро стало понятно, что особых проблем-то и нет. Да, это не простой бинарный формат, но и не самый сложный. Нам достаточно быстро удалось создать качественный универсальный импортёр, который вполне пригоден для повседневного использования.

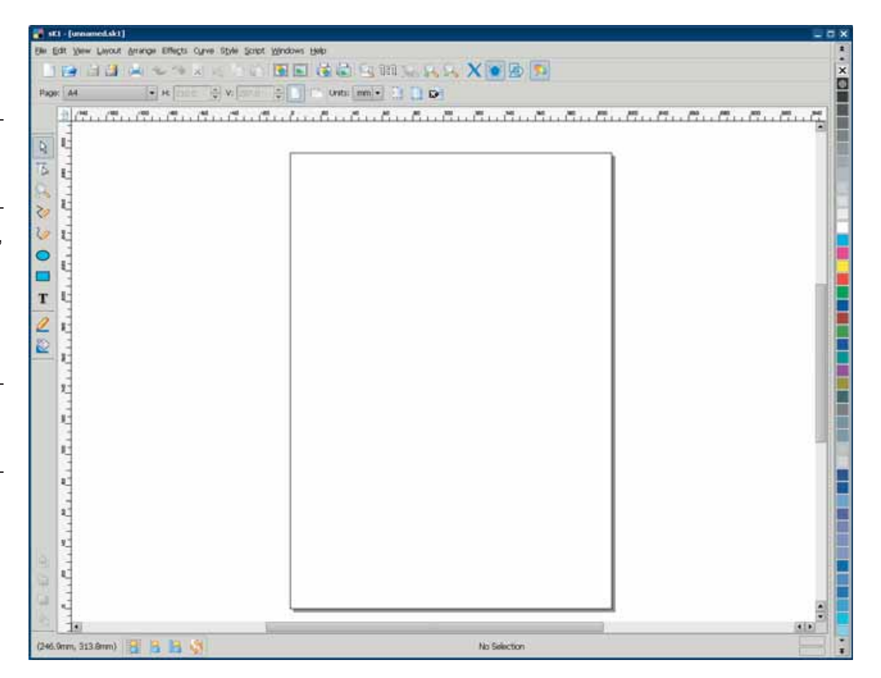
В мае 2007 года *sK1* был представлен на конференции *Libre Graphics Meeting 2007*, в Монреале (Канада). Поскольку импортёр CDR вызвал интерес у других проектов, то сразу же после конференции мы организовали подпроект *UniConvertor* (**http://sk1project.org/**). Это консольное приложение, которое использует движок *sK1* для преобразования файлов между векторными форматами, т.е. выполняет функции **cdr2svg**, **ai2sk1**, **cmx2svg** и т.д. Фактически, *UniConvertor* стал заменой свёрнутому проекту *VectorSection* (*UberConvertor*) фирмы *Xara*, а *sK1*, в каком-то смысле, можно назвать заменой замороженной свободной версии редактора *Xara LX*. Разработчики *Inkscape* уже добавили возможность использования *UniConvertor* для импорта файлов *CorelDRAW* в SVN-версию своего приложения; эта функция также будет доступна в *Inkscape 0.46*.

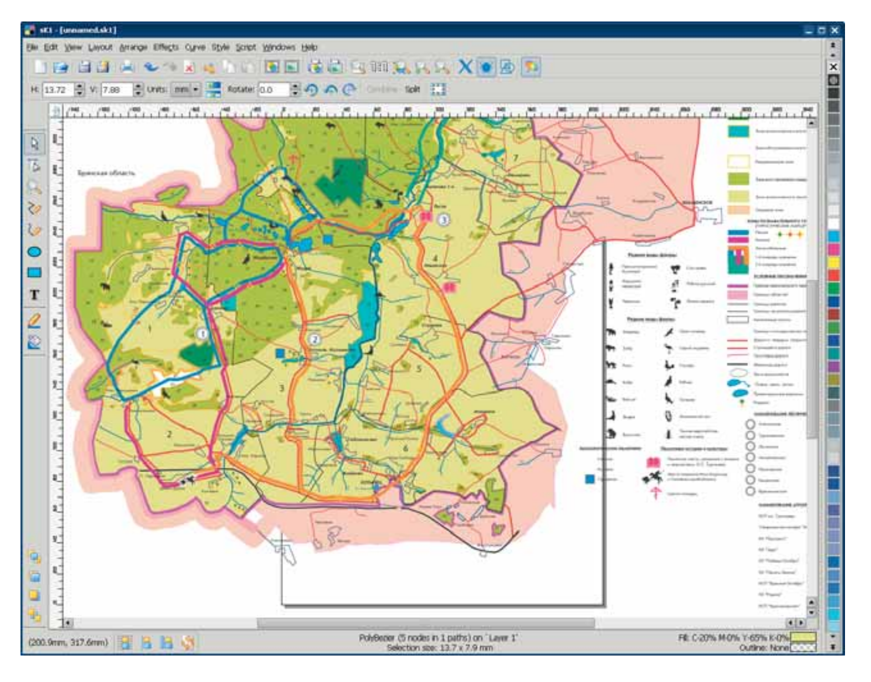
Мы разобрали и добавили поддержку CDT, CCX, CDRX. В результате, на данный момент *sK1* и *UniConvertor* охватывают практически все векторные форматы *CorelDRAW*. Естественно, далеко не все элементы формата поддерживаются полностью, но уже на данном этапе фильтры импорта вполне пригодны для ежедневной работы в полиграфии. Наш CDR-импортёр поддерживает файлы с 7-й по 13-ю версию, сравнимыми возможностями обладает только импортёр *Adobe Illustrator* (поддерживаются версии с 5-й по 10-ю). Коммерческая версия редактора *Xara*, например, может импортировать CDR-файлы версий 3 и 4, что уже давно не отвечает реалиям жизни, т.к. в этих версиях формата нельзя сохранить документы из последних версий *CorelDRAW*.

В октябре 2007 года *sK1* прошёл первичный отбор проектов в конкурсе Les Trophées du Libre и занял второе место в финальном конкурсе в группе **Multimedia/Games**. Конкурс проходил 29–30 ноября в Суассоне, Франция.

LXF: Как распределены обязанности в вашей группе?

ВФ: С моей точки зрения, у нас имеется своеобразный «социальный контракт»: я помогаю расколдовать формат CDR, а Игорь добавляет





в *sK1* (а теперь уже и в *UniConvertor*) поддержку VSD/VSS (форматы *Microsoft Visio*).

LXF: Когда появится стабильная версия *sK1*?

ИН: Стабильная версия 0.9 выйдет после того, как будет закончен шрифтовой движок. Уже сейчас он практически рабочий, остались небольшие нюансы. Возможно, это случится в ближайшие пару месяцев [быть может, уже к тому моменту, когда вы будете читать этот номер, – *прим. ред.*].

А в более длительной перспективе – думаю, что процесс совершенствования программы никогда не закончится. В полиграфической отрасли уйма разных направлений, и каждое из них требует массу специфических «апплетов».

LXF: Ну а что вы ожидаете получить в итоге?

ИН: В нашем понимании *sK1* – это базовая платформа а-ля *Eclipse*, поверх которой можно наращивать различный функционал.

Планы очень просты – довести продукт до приемлемого состояния и вытеснить из базового рынка проприетарные аналоги (улыбается). Всё очень скромно.

Относительно ближайших планов: у нас почти созрели улучшенные импортёры WMF (Windows Metafile) и EMF (Enhanced Metafile). Проведено сравнительное исследование коммерческих и открытых проектов по поддержке формата WMF – скоро этот отчёт будет опубликован на нашем сайте (http://sk1project.org/). На подходе долгожданный импортёр VSD. Выполнены существенные наработки в плане дальнейшего разбора CDR-формата. Реализация этих знаний в коде также планируется на ближайшее время. Начат первичный разбор формата *FreeHand*. Планируется также поддержка режущих плоттеров. В общем, фронт работ достаточно широк.

LXF: На кого вы ориентируетесь при разработке: на профессиональную аудиторию или на «сферического коня в вакууме» — простого пользователя?

ИН: Не секрет, что подавляющее число пользователей таких приложений отнюдь не суперпрофи. Так уж повелось: если надо нарисовать простенькую диаграмму, то берётся профессиональная программа, отредактировать фото – *Photoshop CS3* с уймой неиспользуемого функционала, написать «Hello world» – IDE уровня предприятия и не ниже [*улыбается*].

Именно поэтому первичная аудитория, на которую нацелен *sK1* – это труженики препресс-индустрии. Эти люди хотят мигрировать на Open Source, но им нужен реально работающий инструмент, а не макет.

Для профессионалов же должен быть обеспечен необходимый минимум, и этого минимума с головой хватит «обычным пользователям». Если им потребуется настройка цветов по CMYK, или цветопроба, или registration color (что встречается крайне и крайне редко), то они будут иметь такую возможность. В большинстве случаев, эти свойства не востребованы, и обычному пользователю за глаза хватает *Dia* и *Inkscape*, но психологически это очень важный момент.

Иными словами, мы хотим создать продукт с низким порогом вхождения, как у *CorelDRAW*, и с профессиональным качеством конечного результата (PostScript, который выводится на принтер), как у *Adobe Illustrator*.

LXF: Расскажите, пожалуйста, о *Les Trophées du libre 2007*.

ИН: Конкурс проходил в два этапа. На первой стадии проекты самостоятельно регистрировались на сайте конкурса (http://www.tropheesdulibre.org/?lang=en), и после окончания этой процедуры жюри выбрало из массы проектов по три финалиста в каждой категории. Они должны были прибыть во Францию и представить свои проекты в очном порядке.

Следует отметить, что уровень организации конкурса весьма высок. CETRIL (Centre Européen de Transfert et de Recherche en Informatique Libre), под патронатом которой проходил конкурс, постаралась как можно детальнее подготовить данное мероприятие, максимально содействуя всем участникам.

Сам конкурс проходил в весьма плотном режиме: весь день 29 ноября был занят презентациями, докладами и т.п. Честно говоря, вечером в гостиницу вернулись все просто выжатые.

Из недостатков – конкурс всё-таки франкоориентированный. Это отмечали многие участники. Организаторы, конечно, постарались обеспечить мероприятие синхронным переводом, но не везде и не всегда это было реально сделать. Мне, например, даже не сказали, что переводчик свободно говорит по-русски. Правда, видимо, их смутил тот факт, что я из Украины. Уже после мероприятия переводчик спросил у меня: «А вы принципиально в Украине не говорите по-русски?» [*улыбается*].

LXF: Судя по вашим блогам, вы активно готовились к этому конкурсу. Что именно дотачивалось в *sK1* «под мероприятие» и что обеспечило вам такой успех?

ИН: Планировалось продемонстрировать шрифтовую систему и (если бы повезло с шрифтами) анонсировать релиз. Но, увы, шрифты оказались более тяжёлой задачей, чем мы предполагали, поэтому из новшеств была подготовлена контекстная панель, которая позволила избавится от уймы мелких диалогов.

Лично я не считаю второе место грандиозным успехом. Незнание французского, на мой взгляд, и обеспечило более низкое место, чем могло быть. Но это мое личное мнение.

А вот выход в финал – это результат всех четырёх лет работы над проектом. Нельзя назвать какое-то конкретное свойство *sK1*, обеспечившее прохождение в финал.

LXF: Почему вы выбрали категорию «Мультимедиа и игры», и почему в этой категории среди победителей не оказалось игр?

ИН: Когда я заполнял заявку на конкурс, то просто использовал классификацию на SourceForge.net. Там векторные редакторы отнесены к мультимедиа. В общем-то, всё просто...

А что касается игр… Было много возмущений по этому поводу среди участников. Большинство считало, что игры надо было ставить в отдельную категорию.

ВФ: Кстати об играх: если посмотреть на все проекты, поданные на конкурс в данной категории, то игр там будет подавляющее большинство. При этом бóльшая часть игр, вероятно, сделана одним лицом, но подана на конкурс раздельно. Возможно, участник считал, что это лотерея, и надеялся, что таким образом увеличит вероятность выиг-

рыша. Почему ни одна игра не была выбрана для финала, это, скорее, вопрос к жюри [*улыбается*].

LXF: Что дало вам участие в конкурсе? Есть ли подобные планы на будущее (скажем, Google Summer of Code)?

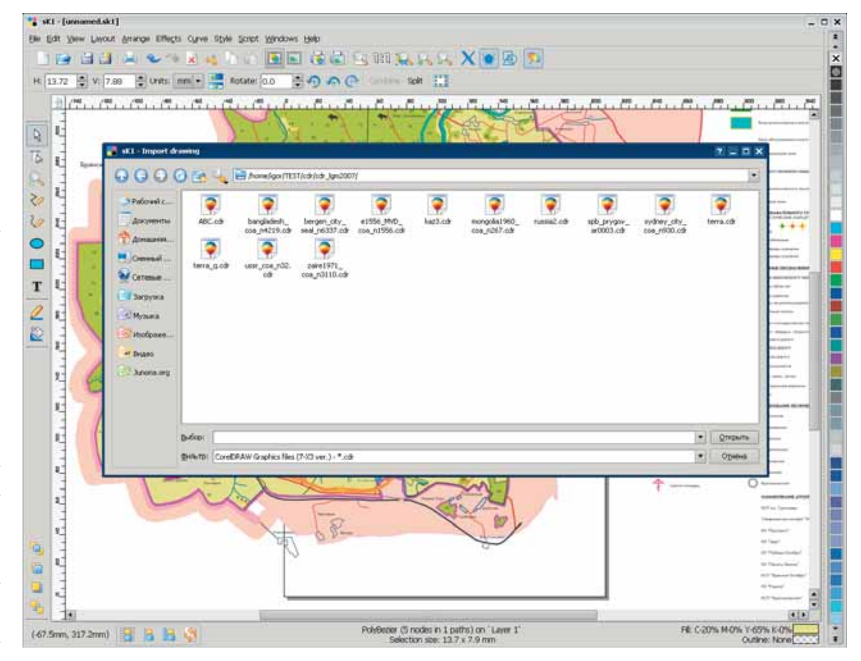
ИН: Думаю, основной результат – известность в кругах Open Source. Что касается будущего, то ближайшее мероприятие – *Libre Graphics Meeting 2008*, которое будет проходить в Бреслау, Польша. Затем летняя конференция GUADEC в Стамбуле. А на *Google Summer of Code* мы не проходим ввиду возрастного ценза – мы давно уже не студенты [*улыбается*].

LXF: Чего, с вашей точки, зрения не хватает сообществу Open Source?

ИН: Прежде всего, не хватает профессионалов в предметной области. В каком-то смысле удачное развитие проекта обусловлено моим многолетним опытом работы в полиграфии. Такой багаж знаний невозможно получить «наскоком по книжкам».

Сходная ситуация и в других областях. Коммерческие проекты могут себе позволить держать и программистов, и предметных специалистов. В Open Source, к сожалению, это затруднено, поскольку при эпизодическом общении тяжело быстро найти общий язык и выработать какую-то стратегию развития проекта. Наиболее оптимальный вариант – это когда программист является специалистом в предметной области. Например, в команде *Scribus* такая же ситуация, как и у нас. В результате проект действительно прочно стоит на ногах.

ВФ: Я бы добавил, что часто не хватает способности договариваться и делать «грязную» работу сообща. Ну и нужно побольше коммуникаторов-информаторов, таких как Александр Прокудин.



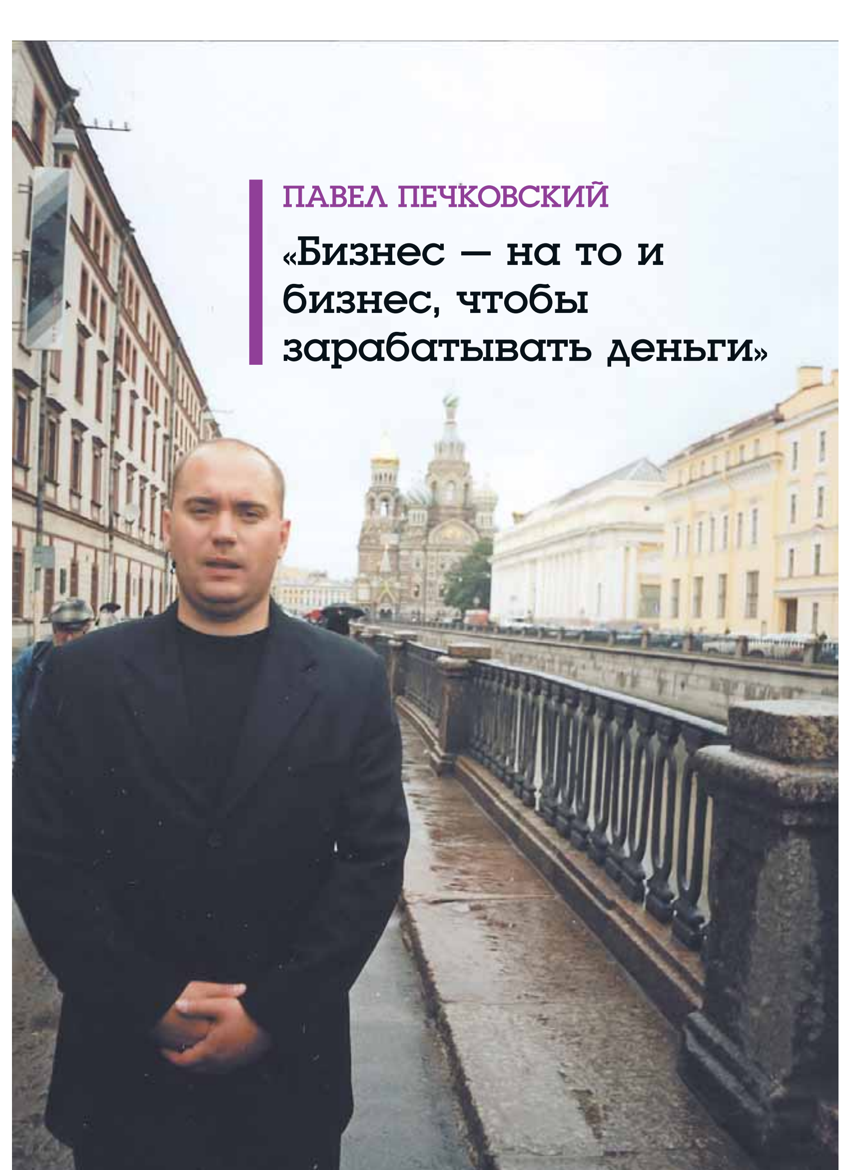
LXF: Чего бы Вы пожелали сообществу?

ВФ: Счастливого Нового Года!

ИН: Хай щастить в Новому Році! LXF

Оптовая торговля с GNU/Linux

Все люди должны чем-то питаться. О том, как еда попадает в магазин, задумываются немногие – но заместитель директора по безопасности ООО «Оптовик» (г. Брянск) **Павел Валерьевич Печковский** знает процесс в деталях. И считает, что Linux в нем – самое место.



ПАВЕЛ ПЕЧКОВСКИЙ

«Бизнес – на то и бизнес, чтобы зарабатывать деньги»

Евгений М. Балдин (LXF): **Какой процент компьютерного парка Вашей компании работает под GNU/Linux?**

Павел В. Печковский (ПП): У нас всего 26 рабочих станций. Из них 22 пользовательских рабочих места уже переведены на GNU/Linux, а к Новому Году к ним присоединятся и оставшиеся четыре ноутбука, которые в данный момент работают на предустановленной ОС Windows Vista Home Basic Edition. Вопросы технического характера и портирования нестандартных приложений уже решены.

LXF: **Что ставится на пользовательские машины?**

ПП: Дистрибутив – *Kubuntu 7.04 Feisty Fawn*. В качестве рабочего стола мы используем KDE 3.5.6, как офисный пакет – *OpenOffice.Org 2.2.0*. Файловые менеджеры – *Krusader 1.80.0* и *Konqueror 3.5.6*. Для работы в Сети применяем связку из *Mozilla Firefox 2.0.0.6*, *Mozilla Thunderbird 1.5.0* и *SIM-IM 0.9.4.2*.

LXF: **О, как подробно! А какой была основная причина перехода?**

ПП: Безопасность. Стабильность. Совокупная стоимость владения (TCO). Как следствие всего вышеперечисленного – конкурентные преимущества компании.

LXF: **Ваша компания занимается оптовыми поставками пищевых продуктов и алкоголя. Последнее я, как убежденный и последовательный трезвенник, не одобряю. Но это, а также то, что пищевая промышленность — весьма прибыльная отрасль, означает, что у компании есть значительные средства. Отсюда следует вопрос: а зачем было переходить на GNU/Linux?**

ПП: Бизнес – на то и бизнес, чтобы зарабатывать деньги. Если не считать денег в самом начале пути, то они никогда и не появятся. Плановый переход на программное обеспечение GNU/Linux не имеет причинно-следственной связи с делом, которым мы занимаемся. Постоянный поиск более эффективных приемов работы и обмен опытом – вот неотъемлемая часть деятельности отдела информационных технологий. В этом же заключается и смысл перехода.

Оптимизация бизнес-процессов и затрат занимает первое место в списке задач ИТ-отдела, и далеко не последнее место среди задач в любой коммерческой организации, которая хочет быть успешной. Это нормально. Это безусловное требование любого бизнеса. Проведя анализ эффективности работы пользователей ИС, мы почитали целе-



› Начальник Отдела продаж алкогольной продукции Наталья Панова.



› Начальник отдела закупок алкогольной продукции Эдуард Купреев.

сообразным использовать на рабочих станциях ПО GNU/Linux. Но говорить о том, что мы решили сэкономить и именно поэтому установили GNU/Linux – это значит ввести в заблуждение читателей. Это не так. GNU/Linux исключительно ради экономии средств – это неправильных подход и неправильная постановка вопроса.

LXF: Составлялся ли план миграции?

ПП: Да, у нас был четкий план миграции. Однако, хочу отметить, что без поддержки руководителя предприятия Е.М.Леоненковой, это было бы невозможно, и никакой – даже самый распрекрасный – план перехода нам бы не помог.

Мы начали сверху – первым перевели руководителя предприятия, и только через две недели начали миграцию отделов и подразделений, от руководства к пользователям. Данный факт обусловлен тем, что поведение пользователей во много обусловлено привычкой, а привязанность к проприетарному ПО в значительной степени обусловлена ленью. Мы этого не приветствуем, так как повторюсь: «мы – коммерческая компания».

LXF: Почему вы остановили свой выбор на Kubuntu?

ПП: Причина в том, что на сегодняшний день, как и год назад, это – объективно лучшее настольное решение, которое можно рассматривать, как альтернативу коммерческому ПО. Кроме того, он основан на Debian – самом стабильном и самом некоммерческом дистрибутиве. Огромное спасибо Сообществу за это.

LXF: Пользуетесь ли вы коммерческой поддержкой?

ПП: Никогда не пользовались и не планируем этого делать, так как существует Сообщество.

LXF: В предварительном разговоре вы упоминали потенциальные проблемы с управлением «К». Насколько эта проблема реальна?

ПП: С учетом вступления в силу Ч.4 ГК РФ с 01/01/2008, проблема более чем реальна.

Ее решением может быть установка в качестве настольной системы лицензионного коммерческого ПО, либо заключение прямых договоров с поставщиками на предмет использования узкопрофильных специализированных продуктов, либо использование ПО на основании лицензий GPL и/или BSD. Насколько это действительно защитит компанию, трудно сказать, по причине отсутствия (на момент взятия интервью) их официального русского перевода. Однако я уверен, что мы живем в правовом государстве, и презумпция невиновности действует в отношении как физических, так и юридических лиц.

Касательно серверных продуктов, впрочем, хотелось бы отметить, что существует ряд субъективных факторов, которые определяют необходимость использования строго определенных решений. В этом случае, мы считаем целесообразным использование только лицензионных, основанных на заключении соглашения между производителем (правообладателем) и конечным пользователем программных продуктов. Таким образом, ряд специализированного учетного ПО в нашей организации является полностью коммерческим. К маршрутизаторам и почтовым серверам это, разумеется, не относится – они распространяются по лицензии BSD.

LXF: Ну, а кто-нибудь выиграл от этого перехода?

ПП: Прежде всего, компания. Чтобы понять почему, рекомендую ознакомиться со следующим документом, находящимся в публичном доступе по адресу http://www.microsoft.com/rus/be-legal/msft_tco_whitepaper.mspx. Там находится отчет об исследовании, проведенном IDC и озаглавленном: «Понимание рисков и затрат компании, связанных с использованием нелицензионного ПО» [спонсором исследования, как нетрудно догадаться, выступает корпорация Microsoft, – *прим. авт.*].

К нашему огромному удивлению, от перехода выиграли и системные администраторы. Телефоны молчат, почта – молчит, корпоративный IM – молчит. Пользователи? Безусловно, так как больше нет проблем, знакомых любой организации, зависящей от Интернета, ведущей бурную электронную переписку, а также использующую транспортные протоколы, общие документы и принтеры. Все работает. Просто все.

LXF: Чего вам сейчас не хватает для полного счастья?

ПП: Того, что, к сожалению, никогда не произойдет – функционирования *1С:Предприятия 7.7.x* в среде GNU/Linux без *WINE*.

LXF: А что бы вы порекомендовали сообществу, чтобы процесс внедрения GNU/Linux в бизнесе пошел активнее?

ПП: Не стоять на месте, не бояться и двигаться вперед, меньше говорить – больше думать и делать. Почаще вспоминать Линкольна: «Я иду медленно, но никогда не двигаюсь назад».

LXF: Традиционный нескромный вопрос: в предварительном разговоре вы упомянули, что журнала Linux Format вам чем-то помог. Чем именно?

ПП: Без него мы не сделали бы то, что сделали, т.е. не было бы никакой миграции. Возможно, даже не начали бы. Он помог нам практическими советами, показал ориентир и поддержал морально, а также дал нам уверенность, что мы не одни. Это важно.

Спасибо вам за то, что вы есть. Вы делаете гораздо больше, чем просто журнал о технологиях. Думаю, этот проект останется в новейшей истории России. LXF



› Александр Сайков, инженер-программист, правая рука Павла Печковского, начальник отдела ИТ — доволен проделанной работой по переходу на открытое программное обеспечение, особенно ее результатами.

Что за штука... Gnome Online Desktop

Увязли в борьбе с бесчисленными окнами и вкладками браузера на своем рабочем столе? Команда Gnome решает вопрос – **Майк Сондерс** уточняет…

Чую, пахнет очередной недоделанной и перегруженной комбинацией Web 2.0 и рабочего стола...

Комбинация – точное слово, завидую тонкости вашего чутья. Но Gnome Online Desktop метит гораздо выше, чем 'push technology', полузабытый проект Microsoft по включению web-страниц в локальную рабочую среду с *Internet Explorer 4.0*. Исторически, web-сайт и рабочий стол плохо уживаются друг с другом – не так-то просто просеять прорву HTML и JavaScript, чтобы найти необходимую информацию. Но теперь, когда сайты Web 2.0 предоставляют API, позволяющие внешним программам собирать информацию и осмысленно ее организовывать, задача упростилась.

А примерчик?

Возьмем вашу учетную запись в Linux. Устанавливая дистрибутив, вы, скорее всего, по-быстрому настроили запись и забыли о ней, не добавив «ничего личного» (портрета и т.п.). Реально ваша индивидуальность проявляется в Сети: на Flickr, Facebook, LinkedIn и даже на eBay, если вы завзятый продавец. Но разве не здорово было бы персонализировать свой рабочий стол? Вообразите, что не нужно запускать браузер, шарить по закладкам в поисках любимого сайта, а то и регистрироваться на нем – всего лишь для просмотра, что новенького в сети. Вместо этого все у вас под рукой, ведь ваш рабочий стол вобрал в себя учетные записи Flickr/Facebook/КакойНиПопадя.com, перестав быть холодным серым миром **login: Вася**. А завершают все это моментальные сообщения от друзей и коллег, рассыпанные на рабочем столе.

> «При удаленном хранении настроек GConf они доступны с других ПК, и ваше ПО будет работать как вам надо.»

Ну да, здорово, но ведь у нас есть апплеты, которые такое умеют?

Да, но это ведь только пристройки к рабочему столу. Апплет, уведомляющий, что ваш приятель выложил новые фотки на Flickr, конечно, полезен, но подобные функции можно глубже интегрировать в рабочий стол. В начале этого года разработчики Gnome вознамерились осуществить это, и родился проект Gnome Online Desktop. Многое в нем пока на стадии планирования (типа «было бы здорово, если…»), но кое-какие части реального кода уже имеются.

Заманчиво. А какие?

Возьмем боковую панель Gnome 'Big Board'. На стадии разработки она выглядит как обычное меню запуска программ, но вместо запуска сугубо локальных приложений оно отображает и ваше состояние онлайн – файлы и музыка общего доступа, ссылки на учетную запись Flickr, страницу LinkedIn и другие онлайн-службы. Здесь можно увидеть, кто из ваших друзей находится в сети службы мгновенных сообщений, каким протоколом он пользуется. Панель может хранить и отображать даже ссылки на программы, которыми вы пользуетесь чаще всего. И если вы переходите с одного Gnome Online Desktop на другой, ничего не меняется, ведь все сведения хранятся в интернете.

И кто же предоставляет эту онлайн-услугу?

Red Hat! Могучий Linux-вендор запустил пока малоизвестную (по крайней мере, по сравнению с Facebook) службу под названием Mugshot (www.mugshot.org). Этот общедоступный сайт с бесплатной регистрацией позволяет каждому собирать в своей учетной записи фрагменты своих любимых сайтов – например, видео с YouTube, закладки с del.icio.us, кусочки Twitter, сообщения Facebook, пункты списка пожеланий (wishlist) Amazon; блоги с Blogger, LiveJournal, MySpace и других блог-сайтов, да много чего еще.

Считайте, это портал: вместо того, чтобы скакать от сайта к сайту, разузнавая, «что там новенького на YouTube? Нет ли сообщений для меня на Facebook? Какие новости на del.icio.us?», вы все это получаете на единственной странице. Более того, этими сведениями можно обмениваться с другими: например, ваши друзья в преддверии вашего дня рождения просмотрят список пожеланий Amazon и купят вам вожделенный подарок (если повезет).

Здорово… Получается, что это все мои интернет-ссылки, собранные в одном месте.

Да, причем интегрированные в рабочий стол. Обычная практика нынешних интернетчиков – включить компьютер, запускать *Gaim/Kopete*, *Firefox/Konqueror* и приступить к регистрации на бесчисленных сайтах. Это неудобно и трудоемко; кроме того, вся работа сосредотачивается в окне браузера, а остальная часть рабочего стола простаивает. С Gnome Online Desktop вы автоматически войдете на все свои сайты и получите список событий, происходящих онлайн – и все это не открывая бесчисленных вкладок в браузере.

Ранее упоминалось, что многие популярные сайты Web 2.0 предоставляют доступ к информации через API (Application Programming Interface, программный интерфейс приложения). В недобрые старые времена, несколько лет назад, едва ли какой-нибудь сайт позволял тянуть с него информацию во внешнюю программу. Единственным способом общения с интернетом было окно браузера. Представьте, что вам надо написать программу, проверяющую сообщения на почтовом сайте: для этого программе пришлось бы прикидываться браузером, входить, обрабатывать HTML и JavaScript, и только потом делать свое дело. И малейшее изменение дизайна сайта зачеркнуло бы всю вашу работу.



Ситуация улучшилась лишь недавно, с изобретением RSS, но и это не идеал. А сегодня многие онлайн-службы позволяют обращаться к ним напрямую – и вы можете написать программу, которая спросит: «Здорово, Facebook, как там насчет сообщений для пользователя X с паролем Y? Привет!». Именно эти новые возможности обусловили появление Gnome Online Desktop.

» Так это только ради моих учетных записей на web-сайтах?

Далеко не только для этого! Как я уже сказал, многое в этой технологии пока на стадии разработки, и, пожалуйста, не сердитесь на меня, если в процессе развития что-нибудь изменится. Но есть идея интеграции с *GConf*, механизмом сохранения настроек Gnome. Например, пусть ваш любимый текстовый редактор – *gEdit*, и вы долго настраивали «под себя» панель инструментов, размер шрифта, подсветку синтаксиса и пр. С переходом на другой Linux-компьютер придется начинать все сначала. А вот Gnome Online Desktop может хранить ваши GConf-настройки удаленно, и на любой машине ваши программы будут работать совершенно одинаково. Круто?

» Э, да мы покушаемся на территорию тонких клиентов! Верной ли дорогой идем, товарищи?

Отчасти мы уже там. Многие из нас пользуются Gmail, Google Documents, Facebook, Flickr и т.п. чаще, чем «нормальным» локальным ПО (*Evolution* или *OpenOffice.org*), и наши настольные компьютеры играют роль тонких клиентов для этих web-сайтов – достаточно иметь приличный браузер уровня *Firefox* или *Konqueror*. Так что тонкие клиенты во многом уже налицо, но мы все еще увязываем их с традиционной концепцией браузера, с закладками и кнопками «вперед-назад». Программисты Gnome Online Desktop не собираются сдавать в утиль обычные приложения: просто они превращают онлайн-службы в элемент рабочего стола, вместо того, чтобы смотреть на мир сквозь амбразуру браузерного окошка.

» И во что все это превратится?

Некоторые детали конструкции (например, упомянутая панель Big Board) уже подгоняются друг к другу. В будущем хакеры Gnome хотят оснастить Gnome Online Desktop типовым механизмом взаимодействия онлайн-служб с рабочим столом. Пример – почта: если у вас Gnome и почтовый ящик на web (Gmail, Yahoo или Hotmail), то, щелкнув на ссылке **mailto:** на web-странице, вы видите, что *Evolution* нехотя запускается и заклинивает вашу работу. Ну и зачем вам гонять локальное приложение за web-почтой?

Зато если бы механизм взаимодействия Gnome Online Desktop знал, как обращаться с Gmail, Yahoo или Hotmail, он открыл бы страничку «Составить сообщение» в отдельном окне браузера. И если из этого окна убрать обычный набор браузерных кнопок, оно почти не будет отличаться от окна «нормального» приложения. Для нас, конечных пользователей, разница становится все менее заметной – и скоро может исчезнуть совсем.

» Кайф! А когда можно будет попробовать?

Прямо сейчас, если вы Gnome-гуру и не боитесь компиляции сырой версии, способной рухнуть в любой момент! Мы же, простые смертные, увидим результаты труда в течение ближайших месяцев: кое-что пробьется уже в Fedora 9. Но наблюдение за процессом развития функций и кода – само по себе удовольствие, и если у вас есть светлые идеи, обращайтесь на **http://live.gnome.org/OnlineDesktop** за подробностями о ходе работы и дальнейших планах. LXF

НОМЕР 100

НОВЫЕ ГОРИЗОНТЫ

Вместе с Майком Сондерсом и некоторыми знаменитостями GNU/Linux вспомним, чего достигло сообщество за 100 последних выпусков Linux Format, и что еще предстоит сделать...

Летающие автомобили, небоскребы пятимильной высоты, постоянные базы на Луне, еда в виде таблеток... Футурологи 1960-х и 1970-х верили, что все это будет в 2000-м году. Люди всегда вглядывались в будущее, чтобы предречь очередной большой прорыв; в большинстве случаев наши догадки не совсем правильны. В глобальном масштабе наше чаяние жить в фантастическом новом мире иногда ставит желаемое выше практичности и исторического опыта.

Зато в мире компьютеров перемены обгоняют предсказания: не дожидаясь отведенных сроков, они часто подносят нам сюрпризы. Ведь, например, в 1996 году компьютеры Apple казались тупиковыми – истощенный нишевый бизнес, без всякого фактора крутизны или возможности выжить в мире, где главенствует Microsoft. Прошло десять лет… и теперь Apple – заново открытая компания, производящая престижные «устройства стиля жизни».

А GNU/Linux? Это самая стремительная и непредсказуемая отрасль из всех. Кто бы мог подумать в начале 90-х, что кучка хакеров свободного ПО, работающих над Unix-подобной ОС для технарей, произведет что-либо значительное в мире компьютеров? Кто бы предположил, что нечто, написанное большей частью любителями, разбросанными по всему миру и работающими в свое личное время, выльется в обеспечение работы миллионов серверов и настольных компьютеров? Кто мог вообразить, что кучка хакеров изготовит продукт, способный двигать Интернетом, реально замахнуться на компьютеризацию стран третьего мира и порождать грандиозный энтузиазм?

Наверное, такого уровня успеха не мог предсказать никто. Но были люди со своим видением и страстью к большим свершениям: разработчики. При всей пылкости сообщества, именно программисты вкалывают, превращая идеи в реальность. В нашем юбилейном, сотом номере *Linux Format* мы заглянем в прошлое и будущее GNU/Linux: куда мы пришли как сообщество, какие крутые вещи на подходе и какие испытания нас ожидают.

Дымящийся GNU

Мы поговорили с некоторыми лидерами мира GNU/Linux. И сейчас, когда мы воркуем над Compiz Fusion, а крупные корпорации переходят на Linux, хотелось бы воздать особую честь и хвалу Ричарду Столлмену [Richard Stallman], чья настойчивость укрепила свободу в недрах нашего сознания. Столлмен начал проект GNU в 1983, с целью дать миру совершенно Свободную операционную систему, а в 1991 году последняя брешь была закрыта ядром Линуса Торвальдса [Linus Torvalds] – Linux'ом.

Именно поэтому 'Linux' иногда называют 'GNU/Linux': отдавая дань проекту GNU, который разработал компилятор, библиотеки и приложения, составившие большую часть нашей ОС. Когда мы встаем на защиту технических преимуществ философии 'Open Source' (надежные приложения, безопасный код), не стоит забывать, насколько важен столлменовский подход 'Free Software', который делает упор прежде всего на социальные преимущества: свободу делиться и изменять программы.

Читайте о прошлом и будущем мира GNU/Linux, и посмотрите, что грядет в 2008. Будущее начинается здесь...



3 мая 2000
» Начало продаж первого номера Linux Format

19 июля 2000
» Sun открывает StarOffice (OpenOffice.org)

25 сентября 2000
» Вышел Red Hat Linux 7

23 октября 2000
» Анонсирован KDE 2.0



Ноябрь 2000
» Сформирована Лига KDE, выборы в GNOME Foundation

6 декабря 2000
» Вышел интернет-пакет Mozilla 0.6

4 января 2001
» Выпущено ядро Linux 2.4

Март 2001
» Анонсирована инициатива Microsoft Shared Source



Ричард М. Столлмен

Основатель проекта GNU и Free Software Foundation

Двадцать четыре года назад я провозгласил план разработки операционной системы, GNU, которая была бы совершенно свободной (уважающей свободу) программой. Неотъемлемыми и определяющими Free Software являются свободы

0 запускать программы по вашему желанию;

1 изучать и менять исходный код, чтобы программа делала то, что вам надо;

2 распространять точные копии; и

3 распространять измененные версии.

Без этих свобод вы под пятой у разработчика.

Прошло пятнадцать лет с тех пор, как комбинация GNU и Linux впервые сделала возможной свободу использования ПК. За это время мы прошли долгий путь. Теперь вы даже можете купить ноутбук с предустановленным GNU/Linux более чем у одного производителя, хотя поставляемые ими системы и не являются Free Software в чистом виде. Так что же удерживает нас от полной победы?

Основным препятствием на пути к триумфу свободного программного обеспечения является инертность общества. Вы наверняка сталкивались со многими ее формами. Множество коммерческих сайтов доступно только под Windows. Программа-наручники *iPlayer* от BBC работает только под Windows. Если вы предпочитаете преходящее удобство свободе, вы будете использовать Windows по этим причинам. Многие компании в настоящее время используют Windows, и недальновидные студенты желают изучать Windows. Microsoft активно поощряет эту инертность, поддерживая учебные заведения, внедряющие зависимость от Windows, и заключая контракты на разработку сайтов, работающих только с *Microsoft Internet Explorer*.

Пару лет назад реклама Microsoft заявила, что Windows дешевле, чем GNU/Linux. Их сравнения были развенчаны, но стоит упомянуть о более глубоком изъяне – их аргументация сводится к инертности общества: «В настоящее время больше технических специалистов знакомы с Windows, чем с GNU/Linux». Люди, дорожащие своей свободой, не пожертвуют ею ради денег, но многие бизнес-руководители считают, что все, чем они владеют, даже свобода, должно продаваться.

Инертность общества состоит из людей, пасующих перед инертностью общества. Уступая инертности общества, вы становитесь ее частью; борясь с ней, вы ее уменьшаете. Мы преодолеваем инертность, разоблачая ее и решаясь не быть ее частью.

Нас тянет назад слабина философии большей части нашего сообщества. Многие пользователи GNU/Linux и слыхом не слыхали об идеях свободы, окрылявших первых разработчиков GNU, и они все еще исходят из точки зрения временного удобства, а не свободы. Этим они позволяют водить себя за нос, через все ту же инертность.

Чтобы изменить это, мы должны говорить о свободе и о Free Software – а не просто о практической пользе, как часто делает Open Source. Так мы можем создать силу и решимость нашего сообщества в противостоянии социальной инертности.



Бернхард Розенкренцер

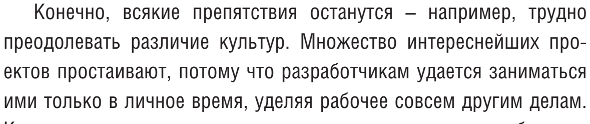
Бывший разработчик Red Hat, ныне основатель Ark Linux

Linux, как и Открытое/Свободное ПО вообще, добился большого прогресса за последние семь лет.

Семь лет назад Linux был хорошей серверной ОС – но не более того. Только-только вышел KDE 2.0 как первая попытка сделать Linux доступным людям, далеким от техники, но отсутствовали многие важные приложения: например, *OpenOffice.org* имелся только в виде смутного кода, и никто не представлял, как его компилировать.

Ныне люди пользуются KDE 3.5 и *OpenOffice.org* как данностью – а проекты, подобные Ark Linux, позволяют каждому установить их. Linux часто используют даже те, кто никогда о нем не слышал – ведь кроме привычных компьютеров, Linux работает и в других устройствах, например, цифровых фотокамерах.

Не за горами выход KDE 4.0, который сделает Linux проще, чем когда бы то ни было, и снабдит разработчиков новыми библиотеками, ускорив и облегчив создание и тестирование новых приложений, и невозможно предсказать, на что будет похож Linux еще через семь лет.



> «Многие бизнес-руководители считают, что все, чем они владеют, даже свобода, должно продаваться.»
> **Ричард М. Столлмен**

Конечно, всякие препятствия останутся – например, трудно преодолевать различие культур. Множество интереснейших проектов простаивают, потому что разработчикам удается заниматься ими только в личное время, уделяя рабочее совсем другим делам. Компании предпочитают не рисковать и остаются в тех областях, где Linux традиционно силен – а реальные инновации происходят там, где семь лет назад никаким Linux'ом и не пахло.



Взгляд назад

Заголовки новостей LXF номер 1:

- » Netscape 6 на подходе
- » SuSE для PPC (Mac)
- » Corel стал «агрессивным» с Linux
- » Консоль Linux – это да! (Indrema)



20 апреля 2001 » Выпущен Mandrake 8.0

1 июня 2001 » Стив Баллмер называет Linux «раком»

18 июня 2001 » Увидел свет GCC 3.0

20 июля 2001 » Первая эмиссия акций MandrakeSoft



16 октября 2001 » Trolltech выпускает Qt 3.0

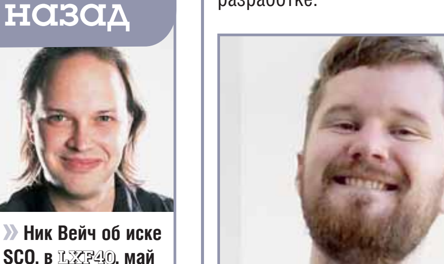
25 ноября 2001 » Начата разработка Linux 2.5.0

Декабрь 2001 » MandrakeSoft рапортует об убытке в 13,5 млн. евро

По технической части, главным препятствием я считаю излишнюю приверженность традиционализму – если что-то всегда работало, это не значит, что лучшего пути нет, особенно для новых задач. Многие проекты, действуя из лучших побуждений, оказывают сообществу медвежью услугу, производя дурное впечатление на новичков. Например, если разработчик видит UI-инструментарий, ради сохранения традиции написанный на C, он, вероятнее всего, с проклятиями убежит обратно на MFC, хотя на Linux есть гораздо лучшие альтернативы.

Другим большим препятствием для Linux является ложное восприятие – которое исчезнет само собой. С точки зрения маркетинга, информация, переходящая от традиционных источников в Интернете – большой союзник Linux'а.

В целом, будущее Linux выглядит ярким – надеюсь видеть рост Linux в следующие семь лет и, конечно, участвовать в грядущей разработке.



Майкл Микс

Разработчик Gnome и OpenOffice.org из Novell

Я воодушевлен тем количеством инноваций (пример: ODF или яркие 3D-эффекты), которые в настоящий момент копируются проприетарным миром. Мы также приходим, наконец, к интероперабельности в бизнес-среде с *Samba*, *OpenChange*, *OpenOffice.org* и *Evolution*.

Мы сдерживаем патенты в Европе; в США, кажется, законы стали немного лучше, и на SCO найдена управа.

Теперь Gartner намекает, что Open Source – главная подрывная технология. Мы можем ожидать еще большего интереса со стороны корпораций. Хотя в этом мы сталкиваемся с теми же угрозами, что и десять лет назад: гарантиями честного лицензирования, прав использования кода, искренности сообщества и хорошим руководством. Open Source обладает мужеством и компетентностью: многие проиграют, победят лишь самые открытые.





Джим Землин

Исполнительный директор Linux Foundation

Я начну с того, где Linux преуспел за время вашего семилетнего существования, и где есть еще место для развития и улучшения.

Linux весьма успешно иллюстрирует идею, что разработка Open Source – лучший способ создавать и продавать программное обеспечение. Linux изобрел новый способ развития приложений и, наряду со многими сегодняшними проектами, вывел разработку за пределы одной компании или страны. Это породило инновации другого уровня, не поддающиеся прогнозированию.

Сегодня Linux обеспечивает работу Google, Amazon, сотовых телефонов Motorola Razr, DVR от Tivo и инициативы One Laptop Per Child. Linux доминирует на рынке телекоммуникаций, никому не уступая по функциональности и эффективности вложения. Linux произвел революцию в суперкомпьютерах. Linux – основной компонент готового недорогого (COTS, cheap off-the-shelf) оборудования в связке со стабильной ОС, которое легко объединяется в сети и распараллеливается, вкладывая в руки правительствам и организациям такие возможности в научных приложениях и вычислениях высокой мощности, которые десять лет назад были им не по средствам.

Linux также удалось заставить Microsoft обороняться. Сейчас Microsoft играет самопровозглашенную роль члена сообщества, но на самом деле она просто стала лучше понимать, как надо атаковать Linux. Не заблуждайтесь, это гонка двух лошадей, и оба соревнующихся выкладываются по полной.

Однако следует отметить, что работы еще очень много. На уровне бизнеса должно происходить реальное сотрудничество. Партнеры Linux должны выбрать – либо они работают над улучшением Linux как целого, и каждый получает больший кусок пирога, либо сосредотачиваются на своем кусочке ради частного успеха. Много внимания уделяется поддержке Linux со стороны независимых поставщиков, Linux на настольных машинах продвинулся на много миль вперед по сравнению с тем, что было 2–3 года назад, но и здесь многое предстоит сделать. Ключевым моментом является стандартизация Linux, и База стандартов Linux (Linux Standard Base, LSB) предназначена обеспечивать обещанную совместимость приложений между дистрибутивами Linux. Все составные части, требующиеся для более широкого применения, уже есть. Нам просто надо производить больше приложений для Linux, для чего требуются стандарты Linux и сотрудничество, как на уровне сообщества, так и на уровне бизнеса.

Что касается 2008 года – я верю, что мы увидим прирост использования Linux в настольных системах. Он будет небольшим,

Взгляд назад

» **Ник Вейч об иске SCO, в LXF40, май 2003:**
«Закон в подобных случаях бывает непредсказуем. Если доходит до суда присяжных, он может быть еще более непредсказуемым. Но не исключено, что суд не входит в планы SCO. Худшим сценарием для них будет явиться туда и проиграть. Может быть, они надеются, что непредсказуемость исхода вынудит IBM отступить. Или надеются на отступные.»



19 января 2002 » Redmond Linux переименован в Lycoris

3 апреля 2002 » Анонсирован KDE 3.0

Март 2002 » Вышли Gtk 2.0, Mandrake 8.2 и Gentoo 1.0

Июнь 2002 » Mozilla 1.0, GNOME 2.0 и Slackware 8.1

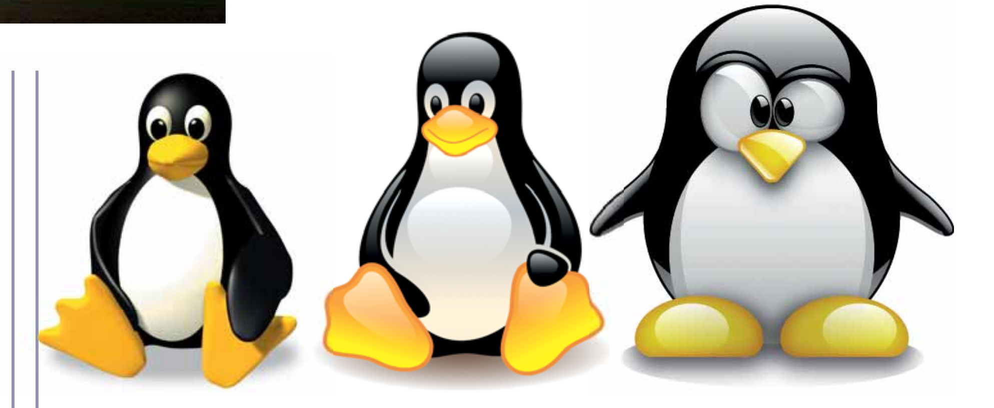


Сентябрь 2002 » Появился Phoenix (позже – Firefox)

16 декабря 2002 » Опубликованы первые лицензии Creative Commons

19 ноября 2002 » Выпущен United Linux 1.0

15 января 2003 » MandrakeSoft объявляет о своем банкротстве



но заметным, и поможет разработчикам сохранить энтузиазм и интерес к расширению границ возможного. Принятие открытых стандартов увеличит использование Linux и Open Source в государственных учреждениях и компаниях, занимающихся внедрением возможностей Web 2.0 в сферы своего бизнеса. Упор на «зеленый Linux» реально уменьшит затраты благодаря снижению потребления энергии системами Linux и увеличению срока службы старого оборудования. Виртуализация приведет к поставкам Linux-серверов. А мобильный Linux увидит еще больше выгоды, так как он дает производителям устройств и провайдерам мобильных сервисов техническую подвижность и меньшее время вывода на рынок. Работа Google в этой области и переход с настольных компьютеров на мобильные устройства протолкнет нас на световые годы в будущее.





ГАЭЛЬ ДЮВАЛЬ

Соснователь MandrakeSoft (ныне Mandriva) и создатель Ulteo

Я считаю, что Linux уже выиграла битву серверов. Кто теперь использует другие ОС на серверах? Да, есть несколько ребят, приверженцев «старой школы», все еще использующих Solaris. Что до остальных, хм-м…. Windows Server 2003? Некоторые сисадмины считают, что плата за Microsoft'овскую клеть пойдет на пользу их карьере, но, кажется, число таковых падает.

Что касается настольных систем, я испытываю смешанные чувства. Трехмерные рабочие столы дали толчок Linux в этой области, но для настольных систем все еще существует большая проблема: на них можно запускать только приложения Linux, а люди ими не пользуются.

Мы живем во все более и более раздробленном и разнородном мире ИТ. Значит, нам нужно дать людям такие решения, чтобы они могли легко связаться с любой частью мира ИТ. В некотором роде, таковым является наш проект Ulteo.



УОРРЕН ВУДФОРД

Возглавляет проект Mepis

Прошлое

В техническом отношении Linux вырос во много раз, но я разочарован тем, что рабочий стол все еще не имеет стандарта. Пока стандарта не будет, разработчики будут неохотно создавать прекрасные новые приложения, основанные на Linux.

У нас есть стандартные приложения и пакеты, например, *OpenOffice.org* и *Mozilla Firefox*. Но все еще существуют две главных системы организации рабочего стола. И у нас есть множество типов приложений, где есть огромный выбор, но нет явного лидера.

У Linux'а есть Линус Торвальдс, главный архитектор и провидец ядра, но нет никого, кто играл бы такую же роль для всей системы, как в области рабочего стола, так и на серверах. Кроме того, OSDL и другие организации, кажется, не заинтересованы в спонсировании того, кто будет делать это. В результате по умолчанию упускаются даже мелкие детали, способные сделать ОС удобнее в работе.

Что касается степени влияния MEPIS, я считаю, что он приносит пользу, демонстрируя другим дистрибутивам некоторые простые вещи, которые можно внедрить, чтобы сделать Linux проще для использования всеми. В доказательство того, что MEPIS оказал некоторое влияние, вспомним, что при выпуске MEPIS 6.5 он был обвинен одним автором в том, что является просто таким же настольным дистрибутивом. Другими словами, множество мелочей, которые первым сделал MEPIS, сейчас являются стандартами, и о MEPIS судят отрицательно.

> «Немногие сисадмины считают, что плата за Microsoft'овскую клетку идет на пользу...»
> **Гаэль Дюваль**

Будущее

Мне кажется, что произойдет вытеснение, и, к сожалению, для нас стандарты Linux станут определяться наиболее коммерчески выгодными Linux'ами, такими как Red Hat, Novell и Canonical. В результате, боюсь, Linux станет посредственным.

Я не думаю, что Linux заменит Microsoft. Сделать это труднее, чем просто создать более качественную операционную систему. Бизнес требует комплексного решения, включающего набор стандартных приложений и популярных компонентов структуры предприятия. Это может произойти, только если этим займется IBM, а это не в ее интересах, поскольку в таком случае компании смогут поменять IBM'овские серверы L Series, и схожее оборудование, на типовую коробку с Linux.

Linux имеет потенциал для домашних настольных систем, но широкое большинство пользователей хочет купить компьютер с предустановленной ОС, а затем забыть про это. Очевидно, они не »

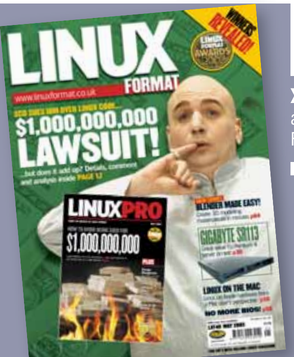
6 марта 2003
» SCO предъявила корпорации IBM иск в размере $1 млрд.

16 июня 2003
» Линус Торвальдс покинул Transmeta и перешел на работу в OSDL



7 апреля 2003
» Вышел Red Hat Linux 9

4 августа 2003
» Novell приобретает Ximian



22 сентября 2003
» Red Hat анонсировал проект Fedora

Октябрь 2003
» OpenOffice.org 1.1, Mandrake 9.2 и SUSE 9.0

Ноябрь 2003
» SCO прислала судебные повестки Торвальдсу и Столлмену

17 декабря 2003
» Выпущено ядро Linux 2.6.0



»

возжелают Microsoft, если у них появится в распоряжении хорошая альтернатива. Но пока крупные изготовители оборудования делали только слабые попытки вбросить на рынок ПК с предустановленным Linux. Я буду с большим интересом наблюдать за Dell и HP в следующем году, чтобы понять, действительно ли они привержены к настольному Linux'у.

Я надеюсь, что сообщество выработает общее мнение, отличное от FSF, которое, мне думается, в большей степени служит раздором в сообществе. И я думаю, что споры в сообществе Linux представляют собой самую большую угрозу будущему Linux'а. Я также надеюсь, что Linux останется независимым и не подверженным денежным интересам. Я говорю, поддержим Linux и Debian!

В 2008 году я буду больше внимания уделять той работе, что мне нравится, которая также дает на хлеб. Помимо MEPIS LLC, мы разрабатываем и поддерживаем Enterprise Architectures and Frameworks. MEPIS Linux будет продолжен, но темпы разработки снизятся. Но теперь, когда MEPIS имеет модель для непрерывного обновления, будет намного проще поддерживать его, а пользователям – иметь самую свежую версию.

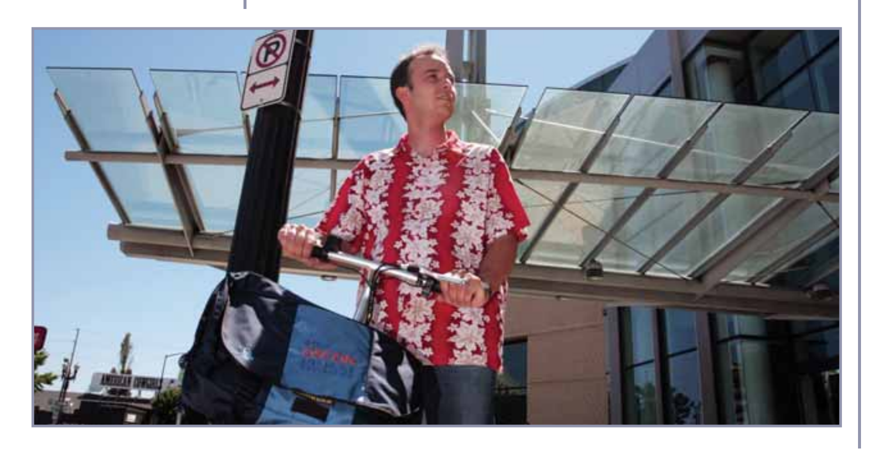


НЭТ ТОРКИНГТОН

Автор и член правления Perl Foundation

Семь лет назад мы находились в спаде. Это было мрачное время для всех в ИТ-индустрии, но оказалось, что это были хорошие времена для Open Source. По мере сокращения бюджетов и средств на ИТ, ранее щедро выдававшихся, люди из окопов обратились к Open Source, чтобы их работа могла быть сделана. Тут-то мы и оценили мудрость «они пришли из-за цены и остались из-за качества», так как когда недавно бюджеты выросли, переход все равно продолжался.

Три самые важные вехи, пройденные за последние семь лет, указывают путь к будущему для разработок Open Source:



(1) **Ubuntu**, сбалансировавший управление сообществом и его вкладами с предсказуемостью релизов. Марк Шаттлворт [Mark Shuttleworth] показал нам третий путь в бизнесе открытых приложений.

(2) **Mozilla**, приносящая Open Source тем людям, которые считают это важным. Мой друг Раэл Дорнфест [Rael Dornfest] (создатель Blosxom) рассказал историю, как он встретил даму, владелицу шоколадной фабрики, одетую в футболку Mozilla, и когда он спросил ее «вы никак пользуетесь Mozilla?», она выдала пылкую и грамотную тираду о том, какая Mozilla великолепная, и почему Open Source сделал ее лучше. Такое мнение потребителя нельзя купить, и оно мостит путь к будущему Open Source, ориентированного на потребителя. (Я умышленно НЕ говорю о «годе Linux на рабочих столах»: это так же больно слышать, как «на сей раз All Blacks выиграют Кубок Мира».)

(3) **Растущее осознание опасности патентов на ПО** для Open Source и для технического развития как программного, так и аппаратного обеспечения в целом. Обнадеживает то, что большая часть проприетарных софтверных компаний также чувствуют, что патенты – скорее экономическая угроза, чем преимущество. Я надеюсь, что в ближайшие годы мы увидим изменения в патентных системах многих стран, вносящие здравый смысл в эту область.

В 2008 я предвижу большое число проектов Open Source, охватывающих кластеры. Надеюсь, что *Mozilla*, *Apache*, Yahoo! или другой друг Open Source предоставит создателям открытых параллельных утилит для тестирования и разработки кластер из 100 и более машин. Ведь Google и IBM уже затеяли некую компьютерную образовательную инициативу с десятком американских университетов. А с наибольшим нетерпением я жду сюрпризов – хочу, чтобы Open Source вытряс нас из технорутины, которая, по моим ощущениям, засосала нас в последние год-два. Побалуйте мой мозг, запустите новую волну крутых идей. Вперед, ребята, я в вас верю!



ЗАК ЭРЛОКЕР

Исполнительный директор по продуктам, MySQL

Лет семь назад — как раз когда начал выходить LXF — я впервые заметил, что Open Source все больше выделяется на общем фоне. *Apache* превратился в web-сервер *де факто*, и Linux, наращивая темпы, становился жизнеспособной платформой для разворачивания приложений, не только web-серверов, но и серверов приложений и баз данных. А разработчики получили весьма надежный инструментарий с скриптовыми языками типа PHP и каркасами типа *Struts* и *Eclipse*.

Хотя и не являясь полностью интегрированным предложением, комплект LAMP (Linux/*Apache*/*MySQL*/PHP) стал реальной альтернативой тяжеловесным разработкам, использующим проприетар-



13 января 2004
» Novell завершает приобретение SUSE

22 января 2004
» Консорциум United Linux прекращает свою работу

Март 2004
» Выходят Mandrake 10.0 и Gentoo 2004.0

Май 2004
» Выходят Fedora Core 2 и SUSE 9.1



16 июня 2004
» Мюнхен переводит 14 000 рабочих мест на Linux

20 октября 2004
» Выпущен Ubuntu 4.10 (Warty Warthog)

Декабрь 2004
» Mozilla рапортует о миллионе загрузок Firefox 1.0

Январь 2005
» IBM передает свободному сообществу 500 патентов







ные программы и языки. Забудьте об утверждении бюджета на дорогостоящие лицензии; просто скачивайте все, что вам нужно, и вперед.

Свободная лесть

Linux Format был там с самого начала, с обзорами продуктов, учебниками, интервью с главными разработчиками и ежемесячным диском на обложке, который помогал распробовать новинки. Я все еще помню свое изумление загрузочным диском Knoppix. Команда *Linux Format* всегда ощущалась как родственные души, работающие вместе, чтобы сделать Open Source доступным более широкой аудитории. Это те парни, с которыми можно сходить за пивом и обсудить заслуги GPL, CVS, KDE, OLPC или дюжину других аббревиатур, и каждый месяц в журнале ощущается этот дух. Поздравляю вас с первыми 100 номерами!

Что касается проектов Open Source, *MySQL* вырос именно за это время. Семь лет назад это была горстка разработчиков, разбросанных по всему миру, работающих у себя дома. Теперь мы – 375 человек в 25 странах, и 90% разработчиков все еще работают у себя дома! *MySQL* – это уже не секретная составляющая знаменитых web-сайтов типа Yahoo, Google и Wikipedia, а база данных общего назначения, используемая в широком диапазоне организаций, малых и больших.

За последние годы мы добавили Enterprise-функции: хранимые процедуры, триггеры, представления и кластеры, что ввело *MySQL* в основное русло. Но мы особо гордимся тем фактом, что *MySQL* все еще быстр и прост в использовании, хоть для профессионального администратора базы данных, хоть для нового разработчика, в первый раз вонзившего зубы в SQL. Мы живем по «правилу 15 минут» – обеспечивая, чтобы пользователь мог за короткое время подключиться и заработать.

Мы продолжаем вводить новые функции в сервер, новые технологии Open Source, наподобие *MySQL Proxy*, новые приложения вроде *MySQL* Workbench и дополнения типа *MySQL* Enterprise Monitor, помогающие администраторам баз данных избегать простоя, сообщая о проблеме до того, как она случится. Наша цель – сделать *MySQL* лучшей базой данных для онлайн-приложений. Модульная архитектура *MySQL* означает, что можно продолжать добавлять новые механизмы хранения – как внутренние разработки типа Falcon, так и от третьих сторон, и делать *MySQL* пригодным для все более широкого класса приложений. А красота Open Source в том, что каждый может принять участие в построении будущего. Чем бы вы ни интересовались в мире Linux, вы можете просто скачать это и браться за дело! »

> «Воодушевляет, что большая часть проприетарных софтверных компаний также чувствуют, что патенты скорее угроза, чем преимущество.»
> **Нэт Торкингтон**

Обратная связь

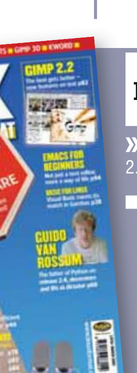


Где, по вашему мнению, Linux играл главную роль за прошедшие семь лет? Чего вы больше всего ожидаете от мира Free Software в 2008? Вы правда думаете, что *GIMP* следует переименовать в *PixelMaster* 3000, чтобы новые пользователи не пугались нехороших ассоциаций [gimp по-англ. «калека»]? Какими бы нелепыми ни были ваши предсказания и предложения, мы рады будем их услышать! Напишите нам на letters@linuxformat.ru или присоединяйтесь к онлайн-дискуссии на www.linuxforum.ru.

Если у вас есть запросы, вопросы, обзоры приложений или темы для учебников, которые вы хотели бы видеть в будущих выпусках журнала, пожалуйста, сообщите нам! Также, если у вас есть какие-либо подсказки, или вы обнаружили, как обойти какую-нибудь проблему Linux, пожалуйста, свяжитесь с нами, чтоб мы могли передать вашу информацию читателям. Подобно движениям Free Software и Open Source, **LXF** приводится в движение сообществом! info@linuxformat.ru

Взгляд назад

» **LXF62** (Янв. 2005) Вердикт по Ubuntu 4.10 в статье *Ultimate Distros*: «Это Debian, только без раздражающего меню Debian, с более новыми, куда лучше упорядоченными пакетами.»



Март 2005 » Выходят GNOME 2.10 и KDE 3.4

Апрель 2005 » Увиден свет Ubuntu 5.04

Июнь 2005 » Выпущены Debian 3.0 и Fedora Core 4

9 августа 2005 » Novell запускает проект OpenSUSE



Сентябрь 2005 » **Первый номер** журнала Linux Format на русском языке

Октябрь 2005 » SUSE 10, Ubuntu 5.10, Mandriva 2006

15 января 2006 » Опубликован первый черновик GPLv3

Март 2006 » Mandriva увольняет своего основателя, Гаэля Дюваля



Выйдет в 2008-м...

Приближается новый год – заглянем в него: какие подарки от Linux готовятся к выходу с конвейера в 2008? Вот некоторые изюминки...



Gnome 2.22

Началась работа над Gnome 2.21, веткой для разработчиков, которая превратится в версию 2.22 (должна выйти 12 марта). На данном этапе команда решает, какие улучшения и изменения следует внести в следующий выпуск: собрано уже с десяток целей. *Ekiga*, Gnome-утилита для VoIP и видеоконференций, будет доведена до версии 3.0, с большими улучшениями: обновленный интерфейс, поддержка большего числа аудио- и видеокодеков через систему модулей расширения, и поддержка IAX2 (протокола, используемого в Asterisk PBX).

В то же время, в *Evince* появится поддержка аннотаций и предпросмотров в выборе файлов, и панель Gnome украсится благодаря новой библиотеке апплетов – Bonobo отдыхает! В недрах Gnome идет глубокая работа, самым заметным результатом которой станет замена GnomeVFS на GIO и GVFS. Это обеспечит работу в связке с FUSE, что в конце концов позволит приложениям Gnome сохранять файлы на учетные записи Google Mail (GmailFS) и на многие другие ресурсы.

Firefox 3

Когда вы читаете эти строки, Firefox 3 (под кодовым именем Gran Paradiso) должен уже существовать в бета-стадии. Хоть Mozilla не установила точного расписания для браузера, но, кажется, при нынешнем темпе разработки мы увидим окончательную версию в самом начале 2008.

Одной из самых важных новых функций *Firefox 3* будут 'Places', комбинированная система Истории и Закладок. Вы сможете отмечать URL звездочкой и пометкой, примерно как в Google Mail, это упростит выбор ваших любимых сайтов и упорядочит их по категориям.

Улучшен менеджер загрузок, появились графические кнопки для приостановки и прекращения загрузки файла; кроме того, теперь имеется поиск в списке текущих загрузок (очень удобно, когда у вас их много, и все выполняются в одно время). По части безопасности, *Firefox 3* предупредит вас, что вы угодили на поддельный фишинговый сайт (например, прикидывающийся сайтом банка) – информация о злоумышленниках берется у Google.



› Ставьте на закладки звездочки и тэги с помощью готовящейся функции 'Places' от *Firefox 3*.

KDE 4.1

KDE'шники по всему миру с нетерпением ждут выхода 4.0, но революции здесь не будет — это скорее способ передать крутые новые технологии рабочего стола в руки разработчиков. Конечный пользователь реально ощутит изменения в последующих релизах 4.x.

Два главных приложения, не успевшие попасть в 4.0 – это *Decibel* и *Akonadi*. Первое дает полную систему связи в реальном времени, связывая обмен мгновенными сообщениями и сервисы VoIP. Со временем оно откроет рабочий стол KDE широкому кругу протоколов связи, построенных по разработке проекта Telepathy. В будущем ожидайте новых великолепных IM-клиентов.

В свою очередь, *Akonadi* – это «PIM storage service». Говоря по-русски, это означает, что авторам приложений не требуется соединять вместе собственную поддержку POP3, IMAP, iCal и других форматов – они могут просто позвать *Akonadi* для этой грязной работы. Это кроссплатформенная разработка, и будет доступна также из приложений Gnome!





Март 2006
› Выходит SUSE Linux 10.1

1 июня 2006
› Выпущен Ubuntu 6.06 LTS с долгосрочной поддержкой

10 августа 2006
› Novell переименовывает дистрибутив для домашних пользователей в OpenSUSE

26 сентября 2006
› Вышла Mandriva 2007



2 октября 2006
› Выпущен Slackware 11.0

15 декабря 2006
› FSF запускает кампанию BadVista.org

8 января 2007
› Fedora Core и Extras сливаются в один проект

4 февраля 2007
› Линус анонсирует ядро 2.6.20



Хрустальный шар LXF

Мы увидели множество перемен за семь лет существования Linux Format. Однако что требуется сделать сообществу в следующие семь лет?



Майк Сондерс

Рынок домашних настольных систем – это крепкий орешек. Для тех, кто проводит большую часть времени в web-браузерах и офисных программах, Linux находится в очень здоровом положении, но большая доля рынка еще не охвачена. Это полупродвинутые пользователи — те, кто играет в самые свежие игры, работает в *Adobe Photoshop* и каждую неделю покупает новую USB-примочку: им нужны приложения с большим именем и драйверы, поддерживающие их новые устройства прямо из коробки.

Мне кажется, что нам также нужно стремиться к стандартам. Свобода – это хорошо, когда выбираешь текстовые процессоры и почтовые клиенты, но кому нужны триллионы приложений по настройке сети? Разве избыток форматов пакетов помогает нам, а не распыляет усилия? Сможем ли мы, наконец, добиться установки программ в явно выделенных каталогах, а не разбрасывать их по всей файловой системе? Я думаю, что чем больше стандартов в основах дистрибутивов Linux, тем легче серьезно осваивать рынок.



Пол Хадсон

Я считаю, что наибольшие опасения вызывает рынок серверов. Да, я знаю, что в этой области Linux'овая смесь акцента на администратора и системы «запусти и забудь» традиционно служат очень хорошо, но это также и та область, где Microsoft и Sun конкурируют особо свирепо, так как она невероятно прибыльная.

Традиционно в области серверов Microsoft играла на замкнутом круге: если вы, например, хоть раз использовали *Active Directory* и *Exchange*, вам будет трудно перейти на что-нибудь другое. А значит, каждый раз, когда кого-то засосала опасная трясина обновлений от Microsoft, ему трудно из нее выбраться. Linux далек от завоевания настольных систем, но в нескольких шагах от серьезных успехов в мире серверов, и я думаю, что наилучший план атаки таков: сделать Linux выбором де-факто для сервера, тогда и победа на настольных системах станет только вопросом времени.



Грэм Моррисон

По-моему, вне главного русла Linux удерживают простые вещи. Недавно я помогал редактору PC Answers устанавливать последний Ubuntu, и тот факт, что человек с высокой квалификацией в Windows просит помощи, вызывает беспокойство. Если уж таким не обойтись без помощи, то большинству пользователей ПК — тем более; а это люди, которых мы должны сейчас обучать Linux.

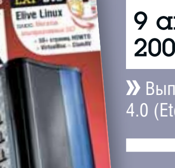
Новые пользователи вынуждены продираться через тернии определения оборудования, проприетарных драйверов и управления пакетами; это неприемлемо. По иронии, наличие выбора, принесшее Linux успех, не дает ему развиваться дальше. Однако для дальнейшего успеха Linux нам нужны стандарты, и это значит, что дистрибутивы должны создать единую концепцию рабочего стола для обычных пользователей. Нам нужен установщик пакетов по щелчку, способный сработаться с любым дистрибутивом; нам нужна пуленепробиваемая настройка и определение оборудования; и нам нужно оставить политику проприетарных драйверов разработчикам и изготовителям оборудования. Пользователи должны видеть цельный рабочий стол, где полнофункциональные приложения просто работают. Мы сделали большой шаг для всего человечества, осталось сделать маленький шажок для одного человека.



Мэтт Нейлон

Самая большая проблема, с которой всегда сталкивался Linux — это преобладающее большинство пользователей компьютеров по всему миру, которые не знают и знать не хотят, что такое ОС. Поскольку им никогда не предоставляли выбор систем, используемых на работе или в школе, им внушают довольствоваться тем, что есть, какую бы ерунду им ни записали на жесткие диски при покупке.

Хотя несколько просвещенных общественных учреждений, например, муниципальный совет Мюнхена и швейцарская полиция, и перешли на Linux, я пессимистично смотрю на прогресс в большем распространении Linux на следующие семь лет – потребуется поколение, чтобы сломать проприетарные кандалы. Как? Хроническое недофинансирование образования по всему миру – это главная мотивация принятия Linux: нищие учебные заведения сначала переключаются на *OpenOffice.org* под Windows, затем переводят всю сеть на Open Source, давая студентам ключ к преимуществам почти на подсознательном уровне. Обидно: люди переходят на Linux не из-за его достоинств, а из-за слабости и ограничений наших закрытых собратьев. Жаль, что самый шумный адвокат Linux'а – Windows Vista. LXF



9 апреля 2007
» Выпущен Debian 4.0 (Etch)

14 мая 2007
» Microsoft заявляет, что Linux нарушает 235 ее патентов

29 июня 2007
» Опубликован финальный вариант GPLv3

10 августа 2007
» Суд США признает Novell, а не SCO, владельцем прав на Unix



18 октября 2007
» Релиз Ubuntu 7.10 (Gusty Gibbon)

Январь 2008
Выходит 100-й номер LXF!



Рокировка с Windows

Изящная альтернатива двойной загрузке: **Джек Найт** объяснит, как превратить реально установленную Windows в виртуальную машину.



Наш эксперт

Джек Найт новое лицо на страницах LXF и технический директор Open Source Migrations. www.osml.co.uk.

Ура, ваш сияющий новый компьютер доставлен, и вы спешите установить на нем Linux. Но вот беда: он зачем-то засорен операционной системой Microsoft Windows! Настала пора сжигать мосты – возврата не будет: отказаться ли от лицензионного соглашения и вступить на тернистый путь тех, кто пытается стрясти обратно свои кровные с OEM-поставщика или дистрибьютора, переформатировать жесткий диск и вытравить все следы Редмондского бегемота; или организовать двойную загрузку, чтобы иногда пользоваться Windows ради приложений, работающих только под этой ОС? А может, она нужна вам для каких-нибудь тестов.

Ну так вот, существует еще и третий путь, к счастью, не имеющий ничего общего с политикой партии: рокируйте их! Linux будет вашей основной ОС, а ваша исходная Windows – виртуальной машиной, работающей поверх него. Более того, у вас даже будет выбор средств виртуализации: лидер продаж *VMware* или мой личный фаворит, *VirtualBox* от Innotek. Решение относительно преимуществ каждого из них я оставляю за вами, так как оба достаточно широко обсуждались и сравнивались. В этой статье мы займемся *VirtualBox*.

BadVista

15 декабря 2006 года Free Software Foundation начала кампанию BadVista, отстаивая свободы пользователей компьютеров и ставя основной целью показ вреда от использования Microsoft Windows Vista и поощрение перехода на альтернативу в лице открытого/свободного ПО. http://badvista.fsf.org



План действий

Мы проделаем следующее:

» Загрузку с Live CD.

» Снятие резервной копии существующего раздела Windows.

» Изменение размера раздела Windows и освобождение места для Linux.

» Установку Linux на освобожденное место.

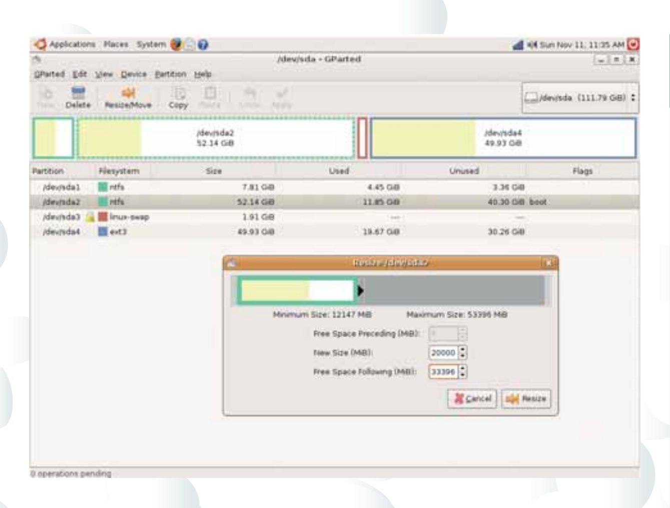
» Конвертацию раздела Windows в VMDK-файл.

» Загрузку Linux, установку *VirtualBox* или *VMware*.

» Конфигурирование VMDK-файла Windows для создания виртуальной машины и ее запуск.

» Установку инструментов виртуальной машины для ее оптимизации.

» После всего этого, если хотите, можете удалить старый раздел Windows для его повторного использования.



› Рис. 1. Убедитесь, что Windows хватит места расправить плечи.

Для выполнения шагов, описываемых на данном уроке, я применил Ubuntu 7.10 (Gutsy), но эти инструкции также должны прекрасно работать на большинстве дистрибутивов Linux, с небольшими модификациями, зависящими в основном от вашей системы управления пакетами. Я собирался взять в качестве гостевой системы Windows XP, но, к несчастью, мой ноутбук с начатой работой сперли в Лондоне, а на его замене оказалась предустановлена Vista. В результате я неожиданно наткнулся на проблемы, и, надеюсь, мой опыт вам пригодится. Windows XP, однако, должна пройти все шаги без каких-либо проблем.

Подготовка

Вспомним, о чем всегда бубнит **LXF**: что надо сделать перед выполнением любых серьезных системных операций на вашей машине? Верно, резервную копию! Так что советую получить доступ к вместительному сетевому или внешнему USB-диску, загрузить вашу машину с современного дистрибутива Linux типа Ubuntu, способного автоматически монтировать большинство USB-устройств и ресурсов *Samba* (NFS требует чуть больше работы), и установить какое-нибудь приложение для резервного копирования. Мне нравится *partimage*: тут можно клонировать существующие разделы посекторно, но с опциями сжатия и возможностью разбиения образа на несколько частей для систем, которые могут хранить максимум 2-ГБ файлы. Для установки *partimage* с Ubuntu Live CD наберите:

```
sudo apt-get install partimage
```

Теперь даже в случае сбоя на следующих шагах вы обязательно сможете восстановить систему.

Изменяем размер диска

А теперь, если он еще не установлен на Live CD, установите и запустите *gparted*, который относительно легко сможет выполнить задачу изменения размера вашего раздела Windows. В терминале наберите:

```
sudo apt-get install gparted
sudo gparted
```

У меня ноутбук с SATA-диском, поэтому соответствующее устройство называется **/dev/sda**. Будь это PATA или более старый тип диска, он был бы **/dev/hda**; короче, у каждого свой путь. Размер статьи не позволяет углубляться в детали переразбиения всех типов дисков, но эта информация свободно доступна в сети.

Порядок действий в любом случае будет таким:

1 Если ваш целевой раздел – NTFS или FAT – смонтирован, размонтируйте его, кликнув правой кнопкой на его пиктограмму в окне *gparted* и выбрав Unmount. Вы можете увидеть не один раздел: OEM-фирмы часто добавляют раздел диагностики или восстановления системы. Лично у меня был раздел, с которого запускался какой-то прохаканый Windows XP для восстановления заводской конфигурации. Я предпочел его не трогать. Нужный вам раздел Windows должен быть больше второго, так что просто подключите свою интуицию.

Подвохи лицензирования

Рекомендуем вам, прежде чем выполнять операции, описанные в этой статье, провести собственные исследования по этому вопросу: похоже, существуют, мягко говоря, «подводные камни», не говоря о различиях в национальных законодательствах, так что сперва убедитесь, что вы поняли ситуацию! Привожу некоторые текущие указания на этот счет, хотя они далеко не всеобъемлющи, и есть определенные правовые «непонятки» на момент написания:

» **Переносимость для виртуальных машин** Microsoft заявила, что клиенты, купившие в розницу ОС Windows XP или Windows Vista, имеют право переносить лицензию на этот экземпляр ОС при переносе (переустановке) его с одного устройства на другое без каких-либо ограничений на число переносов. Однако ПО должно быть удалено с первого устройства перед переносом.

» **Наличие виртуализированной Windows в настольных ПК от OEM-поставщиков** Microsoft заявила, что с внедрением Vista изготовители ПК получили определенные возможности для продажи Windows-клиентов на виртуальных машинах. Неясно, однако, в каких сферах можно их использовать, и какие ограничения применяются к настоящему заявлению.

» **Vista Business / Vista Ultimate** Обе версии Vista, как в коробочной, так и в OEM-поставке, можно запустить в виртуальной машине. Однако имеется целый воз ограничений:

1 Каждая отдельная копия Vista, запускаемая в ВМ, должна иметь собственную лицензию. Установка в виртуальной машине рассматривается как отдельное устройство, поэтому она должна иметь отдельную лицензию.

2 Виртуальные машины нельзя копировать. Вы можете перенести виртуальную машину на другой компьютер, но не сделать копию.

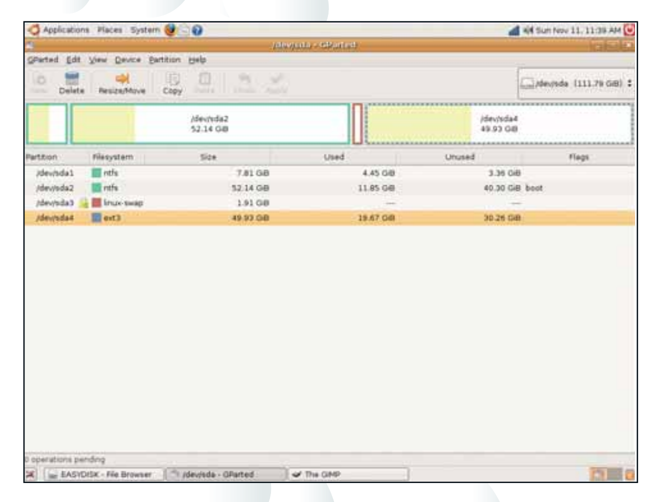
3 Копию Vista от изготовителя ПК, запущенную в ВМ, нельзя копировать или переносить на другую машину. Она лицензирована только для запуска на том же физическом устройстве, что и хост-система, согласно общим ограничениям для OEM-копий Windows.

Последнее ограничение на OEM-копию Vista является одним из самых расплывчатых положений лицензирования. Можно подумать, что сама ВМ считается устройством, но, очевидно, Microsoft вряд ли рассматривает ее как таковое: «устройство», для которого лицензируется OEM-копия – хост-компьютер, а не виртуальная машина. Насколько это юридически чисто и неоспоримо, еще предстоит выяснить.

» **Windows Vista Home и Vista Home Premium** Microsoft не разрешает запускать Vista Home или Vista Home Premium на виртуальных машинах. В этом случае владельцу следует купить розничную лицензию (коробочный продукт) Vista Business, Vista Ultimate или Windows XP для запуска настольной ОС Windows на виртуальной машине.

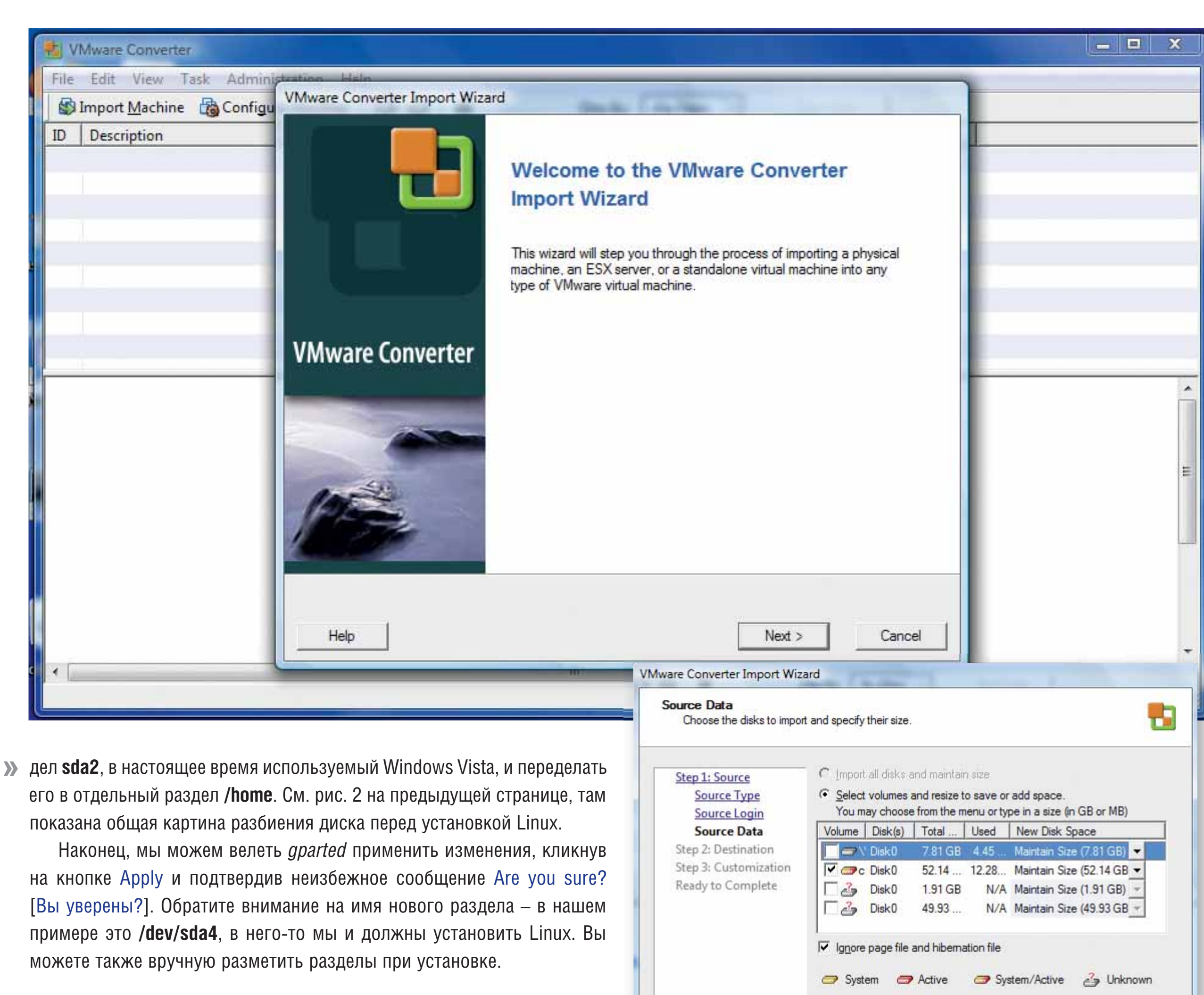
2 Отмонтировав раздел, можете снова кликнуть правой кнопкой мыши на разделе и выбрать Resize. Убедитесь, что вы оставили Windows достаточно жизненного пространства – советую как минимум 10–15 ГБ. См. рис. 1 слева вверху.

3 Далее, кликните правой кнопкой мыши на «неразмеченном» пространстве и создайте раздел подкачки [swap], выбрав тип linux-swap в панели Filesystem. Его размер задайте, исходя из разумного правила: более 2-х объемов ОЗУ [обычно 2.5, – *прим. пер.*]. Затем создайте другой раздел типа EXT3 (выбор по умолчанию) в оставшемся свободном пространстве. Я не создаю отдельный раздел **/home** для себя на этом этапе – хотя всем рекомендую такую конфигурацию как лучшую, и вот почему. После того как я все сделал и запустил, я намерен удалить раз-



› Рис. 2. Разметка диска перед установкой Windows.

»



› **Рис. 3. Кнопка Import Machine запускает процесс.**

› **Рис. 4. Отмените выделение всех томов, кроме тома с Windows.**

» дел **sda2**, в настоящее время используемый Windows Vista, и переделать его в отдельный раздел **/home**. См. рис. 2 на предыдущей странице, там показана общая картина разбиения диска перед установкой Linux.

Наконец, мы можем велеть *gparted* применить изменения, кликнув на кнопке Apply и подтвердив неизбежное сообщение Are you sure? [Вы уверены?]. Обратите внимание на имя нового раздела – в нашем примере это **/dev/sda4**, в него-то мы и должны установить Linux. Вы можете также вручную разметить разделы при установке.

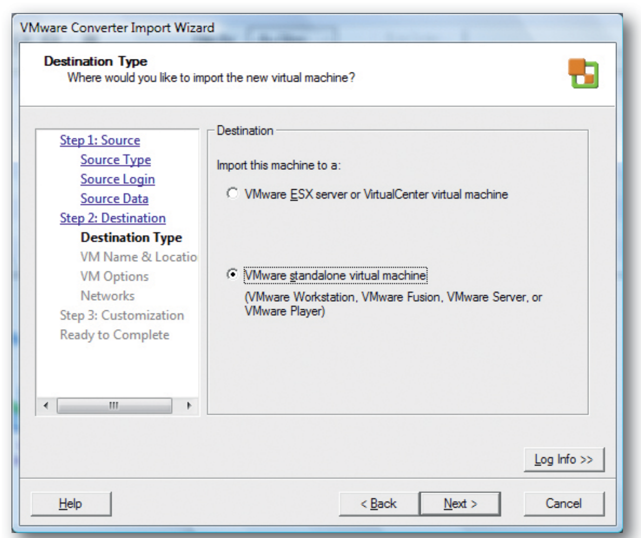
Установка Linux и запуск Windows

Если ваш Live CD имеет опцию установки на жесткий диск, тогда вперед. Иначе вам понадобится перезагрузка.

Теперь загрузитесь с вашего Windows-раздела, выполните все необходимые установки, чтобы все работало, за исключением регистрации копии продукта, так как смена оборудования (на виртуальную машину) может потребовать зарегистрировать ОС снова.

Если вы намерены подключить сетевой диск, чтобы сохранить временную копию раздела, сделайте это сейчас и убедитесь, что у вас достаточно прав для записи в него.

Примечание для Windows Vista: Вам нужно подключить сетевой диск от имени администратора. Самый легкий путь для этого – найти значок командной строки в меню Пуск (или эквиваленте Vista для него – оно уже не называется Пуск). Кликните правой кнопкой мыши на значке и выберите Run as administrator (Запустить с правами администратора). Теперь используйте команду **NET USE** для подключения сетевого диска – вам нужно сделать это, так как Vista не может запустить *VMware Converter* иначе, чем под администратором, и сетевые диски, подключенные обычным пользователем, будут не видны под администратором, и наоборот. Сразу после запуска вы должны разрешить программе выполняться – вот уж воистину полезная функция!

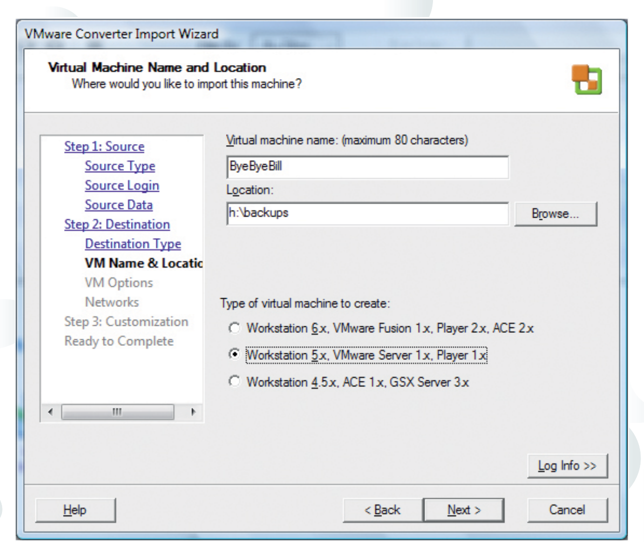


› **Рис. 5. Выберите VMware standalone virtual machine.**

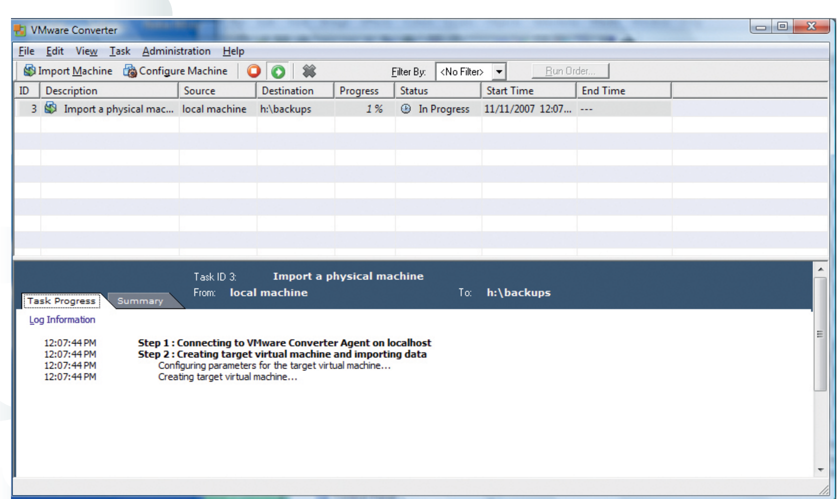
Запустите браузер и скачайте конвертор *VMware* с **www.vmvare.com/download/converter**. Установите его (если у вас Vista, придется постоянно твердить Allow (Разрешить) это, если вы еще не выяснили, как отключить эту опцию!). Перед тем как запустить ее, НАПОМНИМ: если вы используете Windows Vista, вам необходимо КЛИКНУТЬ ПРАВОЙ КНОПКОЙ на значке и выбрать Run as Administrator (Запустить с правами администратора), ДАЖЕ ЕСЛИ ваша учетная запись имеет права администратора! Вам также необходимо дать разрешение на его запуск.

После появления начального окна, как показано на рис. 3, кликните на кнопку Import Machine.

Теперь просто кликайте Next, Next, Physical Computer и отметьте This Local Machine. Затем кликните снова Next, затем ОТМЕНИТЕ ВЫДЕЛЕНИЕ всех томов, за исключением собственного тома Windows, как показано на рис. 4. Здесь вы можете также задать необходимый



› Рис. 6. Разметка вашей виртуальной машины.



› Рис. 7. Превратите вашу установку Windows в ВМ.

вам размер виртуального диска – если вы будете делать это, убедитесь, что оставили гостевой ОС достаточно места для функционирования.

Далее, когда вас спросят о типе назначения (Destination), выберите VMware standalone virtual machine – рис. 5.

Почти все. Задайте имя ВМ – оно произвольное, и введите путь для ее сохранения. ПРИМЕЧАНИЕ: Путь должен указывать не на локальный,

> «Управляйте вашей ВМ Vista изнутри – кликните на значке My Computer.»

а на сетевой диск, упомянутый ранее, или на внешний USB-диск или другой тип устройства хранения данных. На рис. 6 демонстрируется использование сетевого диска. Ранее я подключил его как **H:**. Я также задал тип образа как Workstation 5.x, поскольку твердо знаю, что этот тип работает в *VirtualBox*:

В следующем окне вас спросят о размещении дискового пространства. Я выбрал Allow virtual disks to grow (Позволить виртуальным дискам увеличиваться).

Затем конвертор захочет получить информацию о NIC для ВМ – обойдитесь выбором по умолчанию. В финальном окне, проверьте все заданные опции и нажмите кнопку Finish. Теперь подождите и дайте инструменту закончить свою работу. Я оставил программу работать на ночь.

Большое предупреждение

Имеется пара ограничений, которые необходимо учитывать при установке:

1 В этой версии *VirtualBox (1.5.2)* можно использовать родные образы дисков *VMware* только в р2ежиме 'write through'. Вытекающее из этого ограничение заключается в том, что снимки системы (snapshot) не делаются – и это позор, потому что это очень полезная функция. Если вы хотите сымитировать снимки системы, можно, конечно, сделать резервную копию образа диска, хотя это очень-очень медленно и менее эффективно, чем родная функциональность снимков. Но есть хорошая новость: снятие снимков системы на таких дисковых образах планируется в будущих релизах *VirtualBox*.

2 Если вы используете Windows Vista, не все разрекламированные прибамбасы будут работать на виртуальной машине. Конкретно, не будут доступны эффекты Aero 3D. Но если у вас *Compiz-fusion* или *Beryl*, оно вам надо?

По окончании, мы получим работающий образ, пригодный для использования и в *VMware*, и в *VirtualBox* (который прекрасно работает напрямую с родным форматом *VMware*, **.vmdk**).

Назад, в Linux

Перегрузите машину в ваш новый дистрибутив Linux. Установите *VirtualBox* и скопируйте только что созданный образ диска в какой-нибудь раздел Linux, к которому вы имеете доступ. Я добавлю, что вам необходимо будет установить Guest Additions, иначе все будет тормозить, а дисплей «дергаться» [наличие расширений Guest Additions не влияет на скорость работы ВМ, но обеспечивает ряд полезных возможностей, вроде разделяемых папок, автозахвата курсора мыши и т.п., – *прим. ред.*].

В основных настройках *VirtualBox* нужно будет включить ACPI, IO APIC и VT-X/AMD-V. Я иначе не смог его загрузить. Если хотите звук, не забудьте включить аудиодрайвер ALSA. Вы можете счесть полезным создание общих папок и настроить вашу папку **/home/username**, чтобы можно было читать и писать данные на хост-систему. Я также обнаружил, что сеть прекрасно работает через WLAN, только понадобится NAT.

Другие проблемы Vista в VirtualBox

Дополнительное осложнение с Windows Vista – придется установить специальный сетевой драйвер. Возьмите его с **www.amd.com/us-en/assets/content_type/utilities/V4.51.zip**. Разархивируйте в папку и сделайте файл ISO, который вы сможете смонтировать в *VirtualBox*:

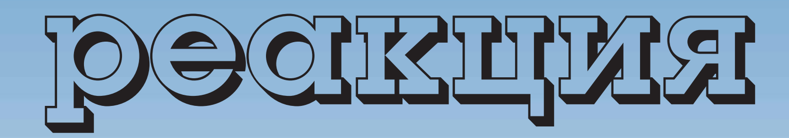
```
unzip V4.51.zip -d vista
mkisofs -o vista.iso -R vista
```

Для его монтирования, в верхнем меню окна *VirtualBox* выберите Devices > Mount CDROM > CD/DVD-ROM Image (Устройства > Подключить CD/DVD-ROM > Образ CD/DVD-ROM) и выберите **vista.iso**, который вы только что создали. Теперь, запустив Vista, кликните правой кнопкой на My Computer и выберите Manage; затем обновите драйвер устройства с CD, который мы смонтировали.

Заключительный шаг

Теперь запускайте вашу новую виртуальную машину. Первый запуск немного затяжной, поскольку требует опознания устройств и настройки. Свершилось! Вы имеете полнофункциональную Windows-машину под контролем Linux, и теперь можете вернуться к части разбиения диска и спокойно удалить оригинальный раздел Windows. Можете использовать его для отдельного раздела **/home** или просто расширить корневую файловую систему, чтоб место зря не пропадало. LXF

Народно-освободительная



Свободная операционная система – это не обязательно Unix. Спешите видеть – Андрей Боровский рассказывает о Microsoft Windows со страниц Linux Format!

Всем известна печальная склонность компании Microsoft ограничивать разработчикам доступ к информации о том, как работают их операционные системы. Самый характерный, на мой взгляд, пример такого рода - весьма оригинальный и очень полезный интерфейс MS-DOS Network Redirector, описанный в книге Undocumented DOS (Schulman et al., 1993). Сколько времени и усилий можно было бы сэкономить, если бы люди из Microsoft опубликовали документацию к этому интерфейсу сразу же после того, как он был добавлен в систему!

Sven B. Schreiber, "Undocumented Windows 2000, Secrets Cookbook"

Вы спрашиваете – возможно ли создать новую Unix-систему за какие-то несколько месяцев, при том, что у вас под рукой нет исходных тестов? Да, это возможно, поскольку для всех важнейших интерфейсов опубликованы общедоступные стандарты.

Erik Raymond, "Samizdat: Stinks on Ice"

В жизни всегда найдется место не только подвигу, но и иронии. Компания Microsoft создала, возможно, самую успешную в мире модель коммерческой разработки ПО и вытеснила коммерческих конкурентов из всех сфер бизнеса, за которые принималась всерьез. Но именно тогда, когда редмондская компания праздновала окончательную победу над OS/2 и Lotus 1-2-3, на сцене появился новый игрок, который пришел из совершенно другой, незнакомой коммерческим гигантам среды. Вскоре новому явлению было придумано и имя – Open Source. С первых дней триумфального шествия движения за свободное ПО в нем выделилось направление, нацеленное на разработку открытых аналогов популярных коммерческих программ, так что нет ничего удивительного в том, что в конце концов на свет появился открытый проект, призванный клонировать самую успешную коммерческую программу всех времен и народов – операционную систему Microsoft Windows.

Собственно говоря, бесплатные аналоги операционных систем Microsoft создавались еще во времена MS DOS. В 1996 году несколько программистов образовали группу под названием FreeWin95. Название проекта раскрывало его цель – создание бесплатного клона стремительно набиравшей популярность 32-битной операционной системы Windows 95. Как это часто случается с амбициозными проектами, группа FreeWin95 распалась, так и не произведя на свет ничего работоспособного. В 1998 году началась разработка нового открытого подобия ОС от Microsoft. На этот раз за основу для подражания была взята не Windows 95, а Windows NT. Сами участники проекта рассматривали свое детище как ответную реакцию на растущий монополизм Microsoft и назвали новую операционную систему соответственно: ReactOS. Между ReactOS и другими проектами-имитаторами операционных систем Microsoft существует одно важное, хотя пока что и не очень заметное, различие: разработчики ReactOS не стремятся клонировать Microsoft Windows. Задача ReactOS – создание открытой системы, совместимой с Windows на уровне приложений и драйверов. В остальном же, говорят разработчики, ReactOS будет развиваться совершенно самостоятельно. Тут, правда, стоит отметить, что поддержка совместимости на уровне драйверов, в том числе устанавливаемых драйверов файловых систем (IFS) требует довольно точного воспроизведения внутренней структуры ядра Windows (которое разработчики ReactOS прилежно изучают).

Хотя сами программисты из команды ReactOS утверждают, что их система является открытым аналогом Windows, так же как Linux является открытым аналогом Unix, между ReactOS и Linux существуют принципиальные различия. На сегодняшний день Unix – это скорее набор спецификаций, нежели операционная система. Спецификации Unix открыты и каждая реализация ОС стремится соответствовать им по мере сил. В случае с Windows эталоном реализации является конкретная операционная система, притом закрытая. В некоторой степени спецификацией Windows можно считать описание API, которое содержится в SDK и DDK, но это описание, во-первых, неполно, а, во-вторых, не всегда соответствует тому, как на самом деле работает система. Поскольку разработчики ReactOS стремятся добиться совместимости с уже существующим закрытым кодом, в случае разногласий между документацией и поведением системы, им приходится придерживаться последнего.

Различия в подходах к доступности исходных текстов системы оказывают на работу программистов даже большее влияние, чем может показаться на первый взгляд. Windows всегда была напичкана различными «секретными» функциями и недокументированными возможностями. Приобщившийся к «тайному знанию» Microsoft (благодаря собственной сообразительности или благоволению компании), получал преимущество над конкурентами. Охота за секретами Windows превратилась в увлекательное занятие и породила весьма обширный

класс литературы (*см. например, процитированную выше книгу Свена Шрейбера*).

Впрочем, разработчики Linux не избежали тех сложностей, с которыми столкнулись их коллеги из мира Windows. Как бы хороша и всеобъемлюща ни была открытая ОС, проблема эмуляции удачливого коммерческого конкурента возникла давно и до сих пор не сходит с повестки дня. Не удивительно, что именно проект WINE (**www.winehq.org**) стал точкой соприкосновения разработчиков Linux и ReactOS. Значительная часть кода WINE может использоваться в ReactOS (и наоборот), и некоторые программисты из группы ReactOS одновременно работают и над WINE.

Знакомство

Операционная система ReactOS (последняя версия – 0.3.3) доступна для скачивания на сайте проекта (**www.reactos.org**) и на **LXFDVD**. На сайте ReactOS красуется картинка с ReactOS CD в фирменной упаковке, но если вы и в правду захотите иметь такой диск, вам придется сделать его самостоятельно. ReactOS доступен в виде образа установочного CD, образа «живого» CD, позволяющего загрузить систему без установки на компьютер или в виде предустановленных пакетов для эмуляторов QEMU и *VMWare* (для знакомства с ReactOS я выбрал последний вариант). Можно, разумеется, собрать систему самому из исходных текстов, причем это совсем не сложно.

Скорость загрузки ОС соответствует ее размерам (то есть, значительно превышает скорость загрузки Windows). В процессе загрузки на экране появляется эмблема ReactOS (**Рис. 1**), затем перед нами предстает рабочий стол, выполненный в классическом стиле Windows 95/NT. На кнопке Пуск вместо эмблемы Microsoft изображена, естественно, эмблема ReactOS. В самом начале работы в глаза бросился неприятный штрих – операционная система предложила установить драйвер какого-то устройства, которое она сама не смогла опознать. Откуда в предустановленной виртуальной среде появилось неизвестное устройство, я не понял, но должен отметить, что окно мастера установки драйверов очень похоже на аналогичное окно Windows. Простим разработчикам мелкие огрехи графического интерфейса вроде меню, которые не всегда вовремя исчезают с поверхности рабочего стола, – все это проблемы роста, которые, я уверен, будут устранены когда разработчики совладают с «начинкой» системы. Структура меню кнопки «Пуск», естественно, напоминает структуру аналогичного меню Windows, хотя и в упрощенном варианте. В меню есть разделы Programs, My Documents и Settings. В разделе игр находим стандартные для Windows «Сапер» и «Солитер». Если ReactOS все еще не подходит для высокопроизводительного офисного труда, то самые популярные убийцы рабочего времени уже на месте (**Рис. 2**).

Говорить о производительности и стабильности системы в сравнении с Windows или Linux, разумеется, еще слишком рано. Пока что о зрелости разработки ReactOS можно судить по количеству поддерживаемых программ. *Firefox* (**Рис. 3**) работает в ReactOS почти без проблем (я, правда, устанавливал его без расширений и Google Toolbar), впрочем, судя по всему, разработчики ReactOS «затачивали» свою систему под этот браузер. *Opera* устанавливается и запускается, но загружать web-страницы, увы, не может. Архиватор *WinRAR* не только установился в системе, но и отлично справлялся со своими обязанностями (следует правда, отметить некоторые огрехи отображения графического интерфейса, имевшие место как в *WinRAR*, так и в неработоспособной *Opera*).

Список приложений, с которыми дружит ReactOS, конечно, не исчерпывается приведенным выше. На сайте проекта можно найти экранные снимки, демонстрирующие запуск *OpenOffice.org* (хотя, поскольку речь идет об эмуляции Windows, демонстрация *Microsoft Office* выглядела бы внушительнее), *Macromedia Flash Player*, *Nero Burning Rome* и даже *Unreal Tournament* (первой версии). Для работы с локальными файлами в ReactOS есть ReactOS *Explorer* и *WineFile*. Есть в системе и свой собственный web-браузер, который пока что бесконечно далек от *IE*. Любопытно отметить наличие в ReactOS четырех рабочих столов, кнопки переключения между которыми расположены на панели задач. Тот, кто подумает, что разработчики ReactOS украли идею нескольких рабочих столов у Linux, ошибется. В Windows эта концепция реализована уже давно, только средства переключения между рабочими столами по умолчанию не установлены.

Майк Сондерс в обзоре *Syllable* (**LXF78**) как-то отметил быстроту загрузки системы. ReactOS тоже загружается очень быстро, но, учитывая небольшое число реализованных системных компонентов, радо-

Уровни документированности Windows API

Клонирование Windows может быть увлекательным делом, в своем роде. Иногда анализ работы системы превращается в детективное расследование. Разумеется, всем программистам приходится искать ошибки в своих (а иногда, и чужих) программах. Эта работа тоже требует некоторых навыков из арсенала Шерлока Холмса. Однако в случае с проприетарными системами детективная аналогия становится еще более очевидной, поскольку программист, изучающий такую систему, часто вынужден искать то, что было вполне сознательно спрятано от его глаз коллегами «с той стороны баррикад». Свен Шрейбер делит интерфейсы программирования Windows на четыре категории, согласно степени документированности:

1 Официально документированные. Описание этих API можно найти в SDK, DDK и MSDN.

2 Полудокументированные. Вы не найдете описания этих API в официальной литературе Microsoft, но входящие в них функции и типы данных можно встретить в общедоступных заголовочных файлах и примерах программ из тех же SDK и DDK.

3 Недокументированные, но не скрытые. Упоминание этих функций и типов данных не встречается ни в литературе Microsoft, ни в доступных исходных текстах, но их можно «выудить» из различных компонентов системы с помощью специальных инструментов (например, отладчика *Kernel Debugger*).

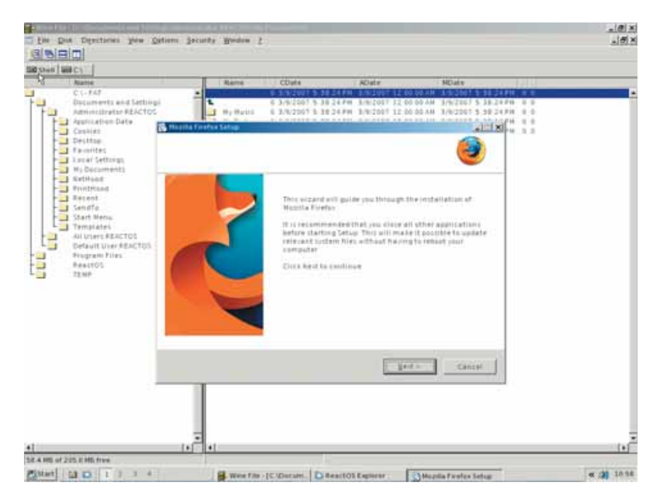
4 Полностью скрытые. Эти API и компоненты так хорошо запрятаны, что обнаружить их можно только с помощью обратной инженерии и инъекции кода. Как правило, скрытые API связаны с очень специфичными механизмами работы Windows, знание которых вряд ли понадобится разработчику эмуляторов наподобие WINE, но может пригодиться тем, кто клонирует ОС Windows (и, конечно, всевозможным злоумышленникам).





› **Рис. 1. (слева) Так стартует *ReactOS*.**

› **Рис. 2. (справа) Наши любимые программы.**



› **Рис. 3. Устанавливаем *FireFox*.**

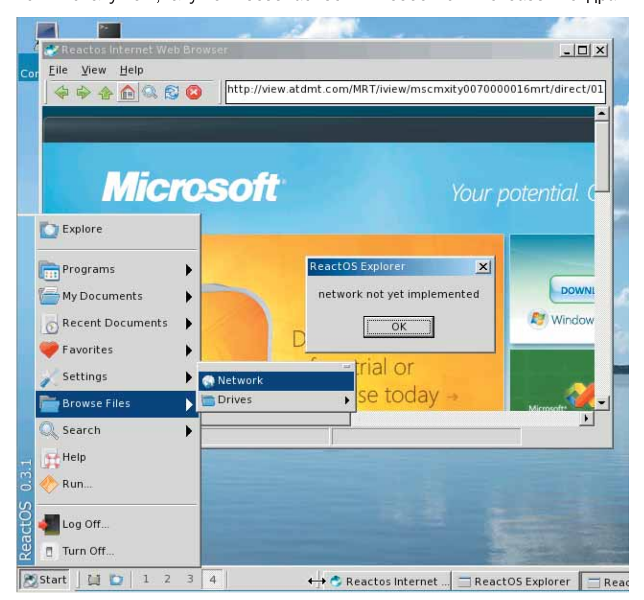
ваться этому преждевременно (то же самое относится и к *Syllable*). Что меня удивило, так это быстрота установки программ из дистрибутивов. По субъективным ощущениям, программы-инсталляторы выполняют свою работу в ReactOS гораздо быстрее, чем в Windows. Вероятно, этот факт объясняется небольшими размерами реестра React OS (не пора ли Microsoft всерьез пересмотреть политику единого реестра)?

Еще один забавный момент, замеченный при работе с ReactOS – система иногда «засыпает» (что заметно по прекращению обновления анимированных элементов и сетевой активности) но может быть пробуждена при помощи перемещения мыши. Вероятно, аппаратные прерывания, посылаемые мышью, активизируют механизм переключения процессов.

Очевидно, что разработчики ReactOS уделяют сейчас большее внимание реализации внутренних компонентов системы. Что касается пользовательского интерфейса, то он станет действительно похож на интерфейс Windows только тогда, когда команда ReactOS реализует свой аналог Internet Explorer: со времен Windows 98 ядро и вспомогательные компоненты браузера Microsoft играют в построении интерфейса пользователя большую роль, чем может показаться на первый взгляд.

Как вы уже поняли, ReactOS свободно чувствует себя в Интернете, а вот работать в локальной сети по протоколам Microsoft не умеет (**Рис. 4**). Это выглядит довольно странно, если учесть, что у разработчиков открытого ПО есть неплохой опыт в реализации этих протоколов. Отсутствует в ReactOS и поддержка NTFS.

Закончим, однако, перечень недостатков, иначе у читателя может сложиться обманчивое впечатление относительно масштабов работы, проделанной командой ReactOS. Как это часто бывает в программировании, самые сложные и интересные вещи спрятаны у системы «под капотом». Такие вещи, как уровень аппаратной абстракции (HAL), системные службы, службы безопасности и собственные базовые драйверы не бросаются в глаза, но разрабатывать их необходимо, а людей, которые могут это делать (да еще в свободное от основной работы время), не так уж и много.



› **Рис. 4. Любителям разделяемых ресурсов придется подождать новой версии.**

Если в мире Linux на вопрос, «какой компилятор использовать», возможен только один ответ, с ReactOS все обстоит сложнее. По вполне понятным причинам разработчики ReactOS используют в качестве базовой платформы Windows (хотя сейчас разрабатывается и среда для сборки ReactOS из-под Linux). Разработчики ReactOS приложили определенные усилия к тому, чтобы в работе над этим открытым проектом можно было использовать открытые (или бесплатные) средства разработки. Благодаря этим усилиям вы можете программировать ReactOS, используя *MS Visual C++ 2005 Express*, *Code::Blocks*, *Borland C++Builder*, *Dev-C++*.

Слово разработчикам ReactOS



Поскольку в настоящее время проект ReactOS выглядит гораздо интереснее изнутри, чем извне, будет более чем уместно предоставить слово самим членам команды разработчиков ReactOS. **Алексей Брагин**, нынешний координатор проекта, любезно согласился ответить на мои вопросы.

LXF: Расскажите о себе. Как и почему вы пришли в проект *ReactOS*?

АБ: Я закончил «Бауманку» по специальности «разработка ПО». Занимался разнообразными проектами, от самых банальных web-приложений до исследований в области обработки видео- и изображений (имеется даже патентная заявка, в США). До ReactOS серьезно увлекался машинной графикой (трехмерные движки, разработка игр – в том числе, на коммерческой основе, в качестве ведущего разработчика). Идея организовать проект по реализации NT-совместимой ОС у меня была довольно давно, и в какой-то момент, когда появилось время, я решил посмотреть, существуют ли уже подобные открытые проекты, и если нет – начать свой. К счастью, их нашлось не много: FreeWin95, ReactOS, FreeDOS, и еще какие-то. Я посмотрел FreeWin95 – он был уже давно заброшен, посмотрел FreeDOS – он не соответствовал моему замыслу, хотя был идейно близок, и наконец я решил взглянуть на проект с непонятным названием ReactOS. После ознакомления с web-сайтом сомнения отпали – это именно то, что я искал. Естественно, я начал по мере возможности участвовать в разработке.

LXF: Как правильнее было бы определить *ReactOS*: как подобие MS Windows или как Windows-совместимую ОС, в которой совместимость с Windows – лишь одна из полезных функций, и которая будет расширять возможности Windows, так что со временем в ней появится функциональность, недоступная пользователям ОС от Microsoft?

АБ: Вся идея и смысл проекта состоит в том, что имея открытую реализацию, любой человек может внести в проект что-то свое, новое. Эта идея движет большинством наших контрибьюторов, ведь одно дело – копировать имеющееся, а совсем другое – сделать это имеющееся лучше. Единственное условие: вся новая функциональность не должна мешать совместимости. Но пока, на данном этапе, главное – достигнуть стабильной работы в основополагающих частях, и не увлекаться различными оптимизациями, улучшениями и тому подобным.

LXF: Представим себе то время, когда *ReactOS* достигнет уровня стабильного продукта. Каковы будут важнейшие преимущества *ReactOS* перед Windows (помимо стоимости дистрибутива) с точки зрения пользователей и программистов?

АБ: Часть ответа уже была в предыдущем вопросе. Важнейшее преимущество – это не цена, а именно открытость, которая позволит реализовать то, чего не хватает в обычном Windows, и то, что там так

мешает. Достаточно почитать любой форум, там есть сотни различных мыслей о том, как сделать Windows лучше. С точки зрения программиста, преимущество есть уже сейчас: вместо чтения не всегда правильной документации MSDN, он может просто посмотреть в исходный код и найти интересующую его функцию, особенности ее реализации. Специально для этого мы стараемся как можно больше комментировать код, предлагать различные тестовые случаи, и т.п.

LXF: Насколько полной должна быть совместимость *ReactOS* и Windows? Ставится ли задача обеспечить поддержку недокументированных возможностей Windows API?

АБ: Настолько полной, насколько это возможно, и это относится к тем недокументированным особенностям, на которые так любят полагаться некоторые разработчики ПО и драйверов.

LXF: Как я понимаю, в целях корректной поддержки драйверов Windows команде *ReactOS* приходится воспроизводить не только API, но и структуру компонентов ядра. Приходится ли вам дизассемблировать модули Windows для того, чтобы понять, как они работают, и воспроизвести их функциональность?

АБ: Мы стараемся максимально избегать дизассемблирования. Есть прекрасные программы, которые позволяют отслеживать вызовы API, собирать текстовую информацию из файлов, есть openrce.org, который предлагает уже готовые цепочки вызовов, которые тоже помогают понять ход алгоритма.

LXF: Microsoft очень ревниво относится к разработкам, которые кажутся им чем-то похожими на их собственные продукты, а в данном случае мы имеем дело с проектом, который сознательно копирует Windows. Как складываются отношения между Microsoft и *ReactOS*, и как, по-вашему, они могут развиваться в будущем?

АБ: Microsoft ревниво относится к тем разработкам, которые снижают уровень дохода корпорации. На сегодняшний день ReactOS – полностью некоммерческий, образовательный проект, и отчасти мы даже способствуем росту приложений и драйверов с открытыми исходными текстами под ОС Windows. С некоторыми разработчиками из команды Microsoft у нас довольно хорошие отношения, и из нашей команды сама Microsoft уже наняла больше трех человек. К сожалению, это означает, что они не могут работать над ReactOS (из-за правовых тонкостей, даже если в Microsoft они работают над совершенно другими вещами).

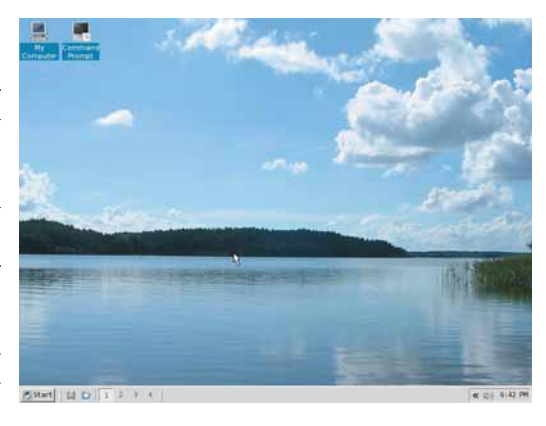
LXF: Большие компании умеют обращать себе на пользу даже неблагоприятные обстоятельства. Хотя *ReactOS* могла бы в перспективе подорвать монополию Windows, не кажется ли вам, что на практике этот проект в некотором смысле укрепляет позиции Microsoft, популяризирует разработку для Windows в сообществе Open Source? Может быть, это и не плохо?

АБ: Сам по себе ReactOS как проект не имеет враждебных намерений по отношению к Microsoft (и уж тем более – к Linux и другим операционным системам). Отдельные разработчики, конечно, имеют свои собственные взгляды, которые не совпадают с официальной позицией.

Что же касается популяризации Open Source-разработки под Windows – это очень хорошо. Открытое ПО должно развиваться как можно шире, захватывать новые области.

LXF: Чему, по Вашему мнению, разработчики Linux могли бы поучиться у *ReactOS* и Windows?

АБ: У ReactOS есть несколько особенностей, которые, на мой взгляд, могли бы быть использованы в других подобных проектах. Во-первых, это унификация и упрощение системы для сборки (ReactOS Build Environment, или сокращенно RosBE). Любой человек может скачать последнюю версию RosBE и исходные тексты проекта, развернуть и набрать "make bootcd". Больше никаких действий – через некоторое время создастся ISO-образ установочного диска ReactOS, собранный полностью из исходников. RosBE существует как для Windows, так и для Linux (в тестовом варианте). Во-вторых, простая установка. Она состоит из трех стадий (разделяемых необходимыми перезагрузками), настройка на конкретное аппаратное обеспечение происходит автоматически, без необходимости ручного конфигурирования. Это также позволяет проводить тестирование более быстро, и даже создавать автоматические системы регресс-тестирования.



LXF: Открытые проекты почти всегда испытывают нехватку «рабочих рук». Кого сейчас больше всего не хватает команде *ReactOS*?

АБ: Больше всего не хватает, конечно же, разработчиков – их, наверное, всегда не хватает. Но мы всегда рады любому другому варианту помощи проекту, каким бы он ни был.

LXF: Что может дать молодому, талантливому программисту участие в таком проекте, как *ReactOS*?

АБ: Знания, опыт, имя. У нас даже в последнее время появилась такая шутка: «Хочешь работать в Microsoft или Apple? Присоединяйся к ReactOS, и никаких проблем при приеме на работу!»

LXF: Каковы ваши личные впечатления от участия в проекте *ReactOS*, оказал ли проект какое-либо влияние на вас?

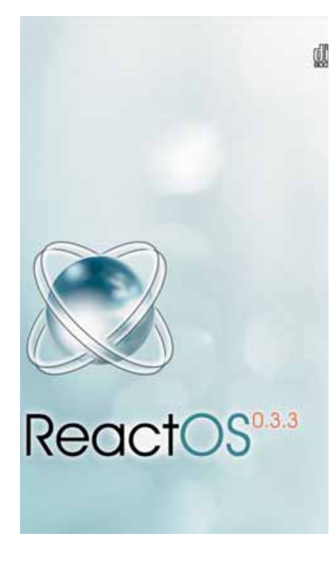
АБ: ReactOS – очень интересный проект. Он дал (и продолжает давать) уникальный опыт, который вряд ли можно получить в другом месте. Даже не в плане программирования (хотя и тут хватает интересных вещей) – это опыт управления крупным специфическим проектом, где вся работа полностью некоммерческая и открыта на условиях лицензии GPL. Эта тема настолько мало раскрыта, что у меня и моих коллег возникает желание написать об этом статью, с названием примерно «Проекты Open Source: особенности управления», чтобы как-то описать полученные знания и наблюдения. Хотя, думаю, случится это не скоро.

Послесловие

Откровенно говоря, я не верю в то, что ReactOS когда-нибудь заменит Windows или Linux на моем компьютере. И дело тут не только в том, что команде ReactOS предстоит еще очень много сделать прежде, чем система станет применимой практически (а ведь другие ОС не будут в это время стоять на месте). Нынешнее благожелательное равнодушие Microsoft по отношению к открытому аналогу Windows объясняется, безусловно, только тем, что ReactOS, ни сейчас, ни в обозримой перспективе, не может стать конкурентом «пожирателя десктопов». Если же когда-нибудь дело дойдет до реальной конкуренции, Microsoft просто обречена на победу. Очевидно, что юридически судьба проекта, клонирующего Windows API, находится в руках разработчика Windows. Сколько технологий, запатентованных Microsoft, вольно или невольно используется в ReactOS? Думаю, что никто этого точно не знает, но при случае редмондская компания найдет, к чему придраться.

Может ли проект ReactOS принести пользу разработчикам Linux? Безусловно, может. От развития ReactOS прежде всего должны выиграть такие проекты, как WINE и Samba. Разработчики ReactOS изучали Windows глубже и, скорее всего, разбираются в устройстве этой ОС лучше, чем другие программисты сообщества Open Source.

В последнее время многие говорят о том, что для завоевания рынка рабочих столов открытым операционным системам потребуется более высокий уровень совместимости с самой популярной коммерческой ОС, чем тот, который предоставляет сегодняшний Linux. Идея открытой операционной системы, максимально совместимой с лучшими коммерческими разработками, весьма привлекательна и, на мой взгляд, перспективна. Хотя такой системой будет, скорее всего, не ReactOS. И, может быть, даже не Linux. LXF





Программировать оборудование? Легко!

Все бывалые линуксоиды умеют писать софт, но чтобы программировать «железо», нужна большая любовь к компьютеру. Мы покажем, с чего начать...

Arduino крут. Крут, поскольку это миниатюрное устройство — примерно три на два дюйма [7x5 см] — имеет порт USB и программируемый чип. Он крут, потому что его можно запрограммировать, используя очень простой язык под названием Wiring. Но главное, вся документация производителя доступна по лицензии Creative Commons, и вы можете собрать устройство сами, если захотите. Правда, для большинства людей это, вероятно, все-таки экстрим, и поэтому продаются также готовые платы Arduino — причем по очень низкой цене. IDE для программирования Arduino доступна по лицензии GPL для множества платформ, и единственное, что стоит между вами и любезным вашему сердцу «железным» проектом — это плата Arduino, клевая идея и кое-какие мелкие детали, аккуратно втыкаемые в Arduino, чтобы сделать его гораздо более интересным.

Электроника и безопасность

Пожалуйста, учтите, что электронные компоненты очень чувствительны: статическое электричество может быть смертельно для вашего Arduino и других крошечных компьютерных частей, так что не шаркайте ногами и носите антистатический браслет. А также помните, что электронные компоненты могут, в свою очередь, быть опасны для вас — они часто содержат свинец, поэтому тщательно мойте руки после работы с вашим прибором!

Приступим к установке

Платы Arduino доступны в нескольких вариантах, но основные три – Arduino NG (“next generation” – «новое поколение»), Arduino NG Plus и Arduino Diecimila. Это не платы-конкуренты – NG появилась первой, потом ее сменила NG Plus, а за ней вышла Diecimila. Мы пользуемся NG, потому что купили ее давно, но сейчас при попытке приобрести ее вы обнаружите только Diecimila. В принципе, различия между этими платами минимальны: на NG Plus установлен процессор Atmega168, на NG – ATmega8, а Diecimila незначительно улучшена для более простой загрузки программ. Чипы ATmega8 и ATmega168 очень похожи; основное отличие в том, что в 168 больше места для программ, но для целей нашего урока это не станет проблемой – подойдет любая из трех.

Где взять плату Arduino? Рекомендуем PCB Europe (**http://pcb-europe.net**) – у них есть в продаже Diecimila за 22 евро (на момент написания статьи), и они с радостью ответят на вопросы, если вы не знаете точно, что вы хотите.

Теперь перейдем к вопросам посложнее: что втыкать в ваш Arduino? Сама плата имеет 14 цифровых разъемов и 6 аналоговых, а также встроенный светодиод и кнопку сброса, и нужно докупить дополнительные детали, чтобы что-то создать. Если вы из США, то, возможно, недалеко от вас есть магазин Radio Shack, где имеются наготове залежи прикольных штучек – просто подойдите к одному из продавцов, объясните ему, что вы пытаетесь сделать свой первый электронный проект, и уйдете с полной корзиной товаров долларов этак на 30. Если вы из Великобритании, то лучшее место разместить заказ на нужные



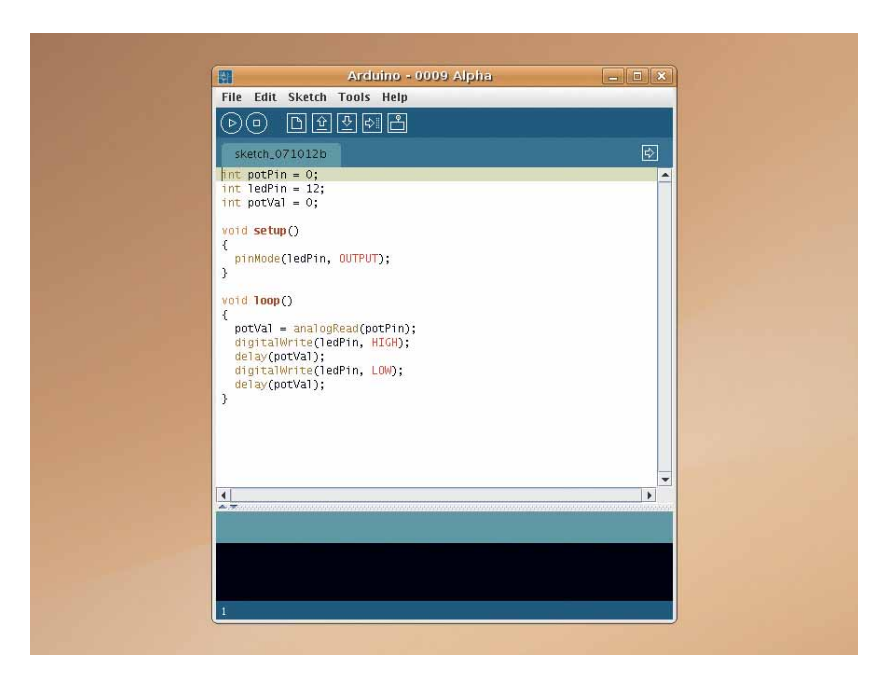
Кабель USB с разъемами типа 2 (слева) **и 1** (справа) **нужен для соединения Arduino с вашим ПК.**

детали – www.maplin.co.uk; также рекомендую прихватить у них один из чудесных наборов Lucky Bags. Поразительно, сколько идей может подсказать случайная смесь деталей! Вне США и Великобритании, либо поищите хороший магазин электроники в вашей стране (говоря «электроника», мы не имеем в виду «продажа iPod»!), либо обратитесь в PCB Europe и купите один из комплектов деталей к Arduino.

Так или иначе, вот необходимый минимум деталей для нашего урока про Arduino:

- **Макетная плата для сборки без пайки.** Обычно это прямоугольник из белого пластика, со множеством отверстий, позволяющих прикрепить детали и соединить их электрически.
- **Несколько проводков-перемычек.** Можно накупить их разной длины, или нарезать самим – без разницы.
- **USB-кабель Type A – Type B.** Говоря по-человечески, это обычный соединительный кабель USB: на одном конце плоский, на другом почти квадратный.
- **Несколько светодиодов.** Лучше – разных цветов!
- **Резисторы разных номиналов.** Сопротивления более 10 кОм не понадобятся, но стоят они копейки, поэтому можете набрать всяких.
- **Потенциометр.**
- **Фоторезистор.**

Вы уложитесь при покупке вышеперечисленного в сумму менее 500 рублей, и даже еще останется, так что не бойтесь набрать лишнего – потом пригодится!



IDE Arduino очень проста: предлагает подсветку кода, сохранение и загрузку и ничего более – зато вы можете загрузить все на вашу плату прямо отсюда.

> «Первая вещь в любом программистском проекте обычно похожа на тест вроде 'hello world'.»

Мигалка

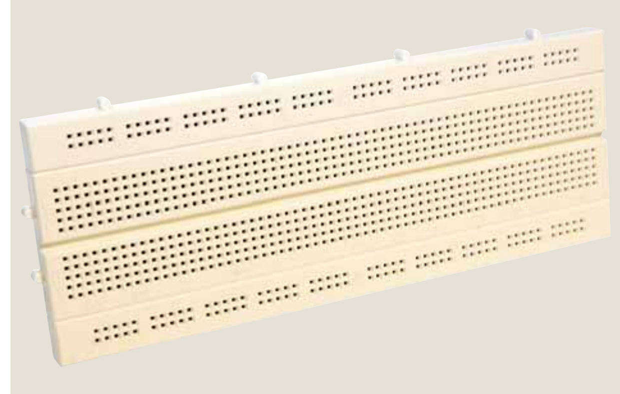
Первое, что делают в любом программистском проекте – аппаратном или каком еще – обычно нечто вроде "hello world": программу заставляют выдать во внешний мир простое сообщение, чтобы убедиться в правильности конфигурации.

В Arduino встроено несколько светодиодов – есть, например, TX- и RX-светодиоды, мигающие при передаче и приеме данных. Мы воспользуемся специальными тестовыми светодиодами: они покажут, что плата нормально работает. Заставить Arduino работать под Linux придется хитростью, потому что она требует Java – причем официальной Java от Sun, клонов она не признает. Если у вас Ubuntu, подключитесь к репозиторию Multiverse, затем установите пакеты *un-java5-jre*, *gcc-avr* и *avr-libc*. Многие пользователи имеют проблемы из-за конфликта Arduino с поддержкой на клавиатуре системы Брайля; если она вам не нужна, сперва удалите пакет *brltty* – а если вы вставили плату Arduino при установленном *brltty*, выньте ее и воткните уже после удаления пакета. Установив программное обеспечение, запустите `sudo update-alternatives --config java` и выберите номер официальной Java от Sun. Этого должно быть достаточно, чтобы все заработало. »

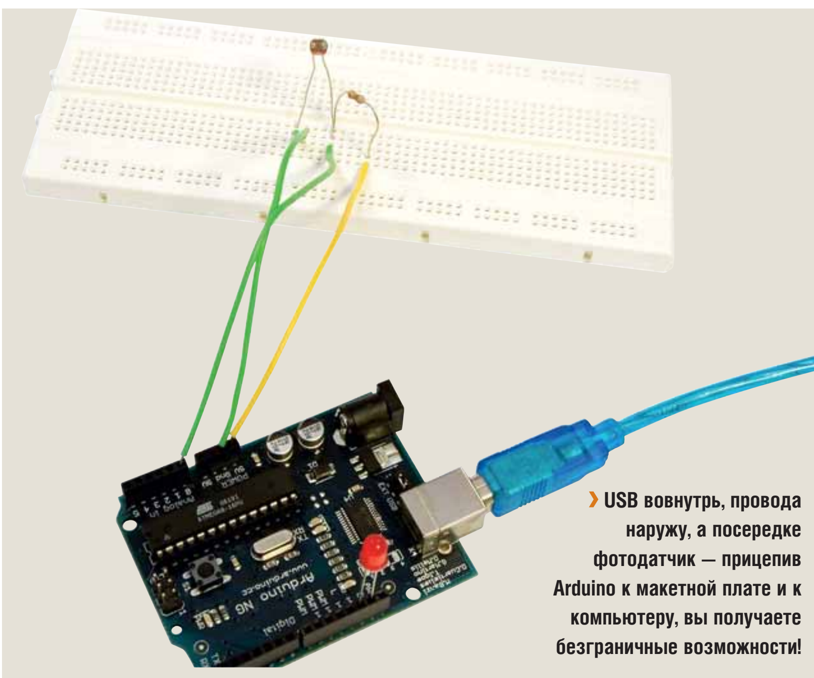


Плата Arduino миниатюрна, и ее можно вставить в самые разные места.





› Полномерная макетная плата, не требующая пайки, обычно имеет около 60 разъемов в ширину и как минимум 10 в высоту. На этом рисунке контакты рядов соединены между собой, а контакты колонок не соединены.



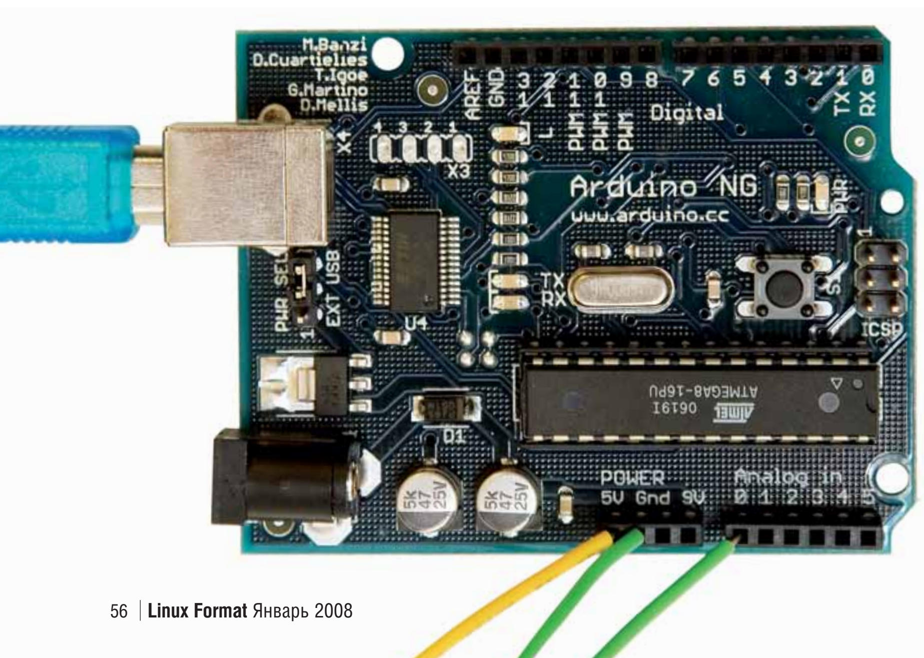
› USB вовнутрь, провода наружу, а посередке фотодатчик — прицепив Arduino к макетной плате и к компьютеру, вы получаете безграничные возможности!

» Если у вас не Ubuntu и не другая система на базе Debian, то вам опять же потребуются Java от Sun, *gcc-avr* и *avr-libc*, но зато в таких дистрибутивах, скорее всего, не придется беспокоиться об *update-alternatives*.

Мы включили IDE Arduino на DVD этого месяца – распакуйте ее на ваш рабочий стол и запустите команду arduino. При наличии проблем попробуйте запустить ее из терминала и посмотреть сообщения об ошибках. Запуская IDE впервые, вы должны будете ответить на вопрос о месте сохранения ваших программ – поддиректория в вашем домашнем каталоге вполне подойдет.

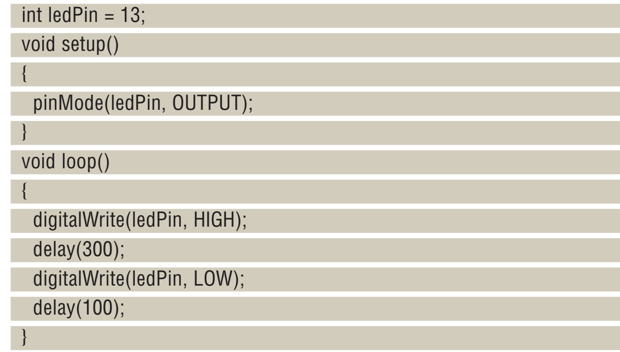
Итак, ПО для Arduino установлено; теперь подключим Arduino к компьютеру через USB-кабель. Если плата работает нормально, светодиод PWR (power, питание) будет гореть, а тестовый диод – мигать, показывая, что на плате все в порядке.

› Главное на плате Arduino — чип ATmega, но видны также аналоговые разъемы вверху и цифровые — внизу.



В редакторе Arduino зайдите в File > New для ввода нового кода. Файлы кода в Ардуинии называются “sketches” («наброски»), и с ними очень легко начать работать. Начнем мы с простого проекта: будем переключать светодиод из ВКЛ в ВЫКЛ, и я хочу показать вам код и убедиться, что он работает, а потом разобраться, как он работает.

Вот этот «набросок»:



```
int ledPin = 13;
void setup()
{
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  digitalWrite(ledPin, HIGH);
  delay(300);
  digitalWrite(ledPin, LOW);
  delay(100);
}
```

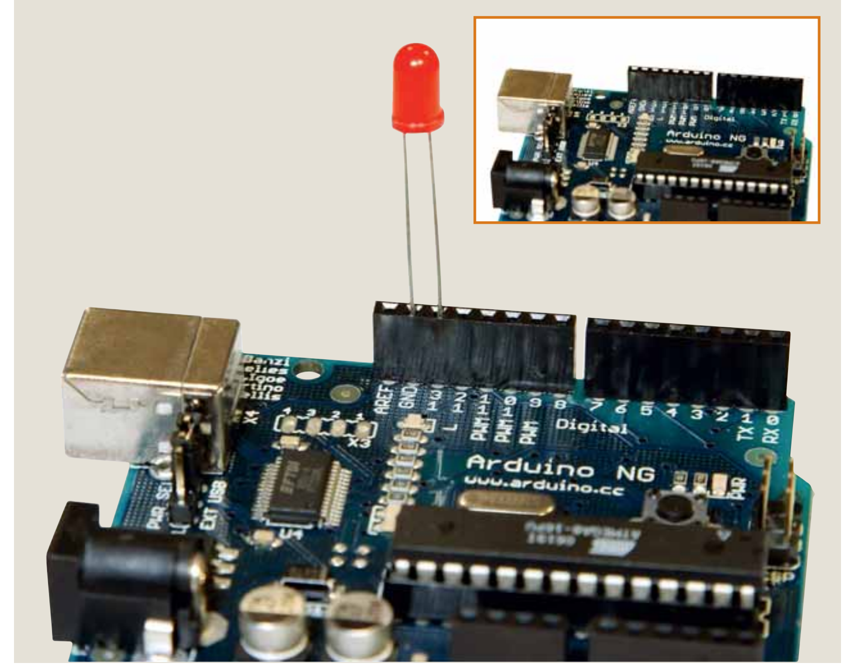
Теперь идем в Tools > Microcontroller и выбираем либо atmega8 (если у вас ArduinoNG), либо atmega168 (если у вас Diecimila). Если вы этого толком не знаете, пропустите данный шаг – Arduino определит процессор при подключении и выдаст ошибку, если обнаружит неправильный. Получив сообщение об ошибке, просто смените опцию! Когда процессор сконфигурируется, зайдите в Tools > Serial Port; вы увидите список возможных USB-устройств. В Linux, вероятно, надо будет выбрать /dev/ttyUSB0.



› В моей макетной плате ряды обозначаются цифрами, а колонки — буквами. Я так на них и ссылаюсь, чтобы вы смогли в точности скопировать мою схему.



› При повороте ручки потенциометр возвращает большее аналоговое значение — убедитесь, что используете analogRead()!



OK, настройка закончена – пора выгружать ваш «набросок» на плату! В панели меню IDE, вы можете увидеть иконку со стрелкой вправо – это кнопка Upload [Выгрузка]. Для Diecimila вам достаточно нажать на кнопку прямо сейчас, и программа будет выгружена. В противном случае сперва нужно нажать на маленькую кнопку сброса на плате Arduino, подготовив ее к принятию нового «наброска», а затем нажать кнопку выгрузки. Светодиоды TX и RX секунду поморгают, затем секунд пять ничего не будет происходить (Arduino ждет, не поступят ли дальнейшие инструкции), и наконец ваш тестовый диод должен начать мигать. Заработало!

С этим маленьким проектом вы можете шагнуть дальше. Возьмите из набора светодиод и подключите его к цифровому контакту 13. Если

> «Если светодиод вставлен неправильно, не беда: просто переверните его.»

вы раньше со светодиодами не работали, хорошенько рассмотрите его перед установкой: у него две ножки, причем разной длины, и одна сторона цветного пластикового ободка плоская. Таким образом указывается полярность: ножка покороче и плоский обод – это «минус». Подключая ваш светодиод к плате Arduino, убедитесь, что плюсовая ножка подключена к контакту 13, а минусовая – к Gnd (ground, земля), и если все будет нормально, вы увидите, как он мигает в такт тестовому светодиоду. Ошиблись – ничего страшного, просто побыстрее выдерните светодиод!

Как работает код

Теперь, когда плата Arduino работает правильно, объясню вам, как работает код, чтобы вы сами могли его модифицировать:

» **int ledPin = 13** описывает переменную ledPin типа integer (целое число – ну, то есть, не вида 3,1) и присваивает ей начальное значение 13.

» **setup()** – функция по умолчанию в Arduino. Она вызывается при запуске вашей программы, чтобы вы могли задать начальную конфигурацию.

» **pinMode(ledPin, OUTPUT)** сообщает Arduino, что вы хотите посылать данные на контакт 13, а не считывать их.

» **loop()** – еще одна функция по умолчанию Arduino: она вызвается всякий раз, когда процессор ищет, что бы ему еще сделать.

» **digitalWrite(ledPin, HIGH)** означает «послать значение HIGH на контакт 13»; HIGH – эквивалент двоичной единицы против двоичного 0, оно же – «ВКЛ», в отличие от «ВЫКЛ». Это включает светодиод.

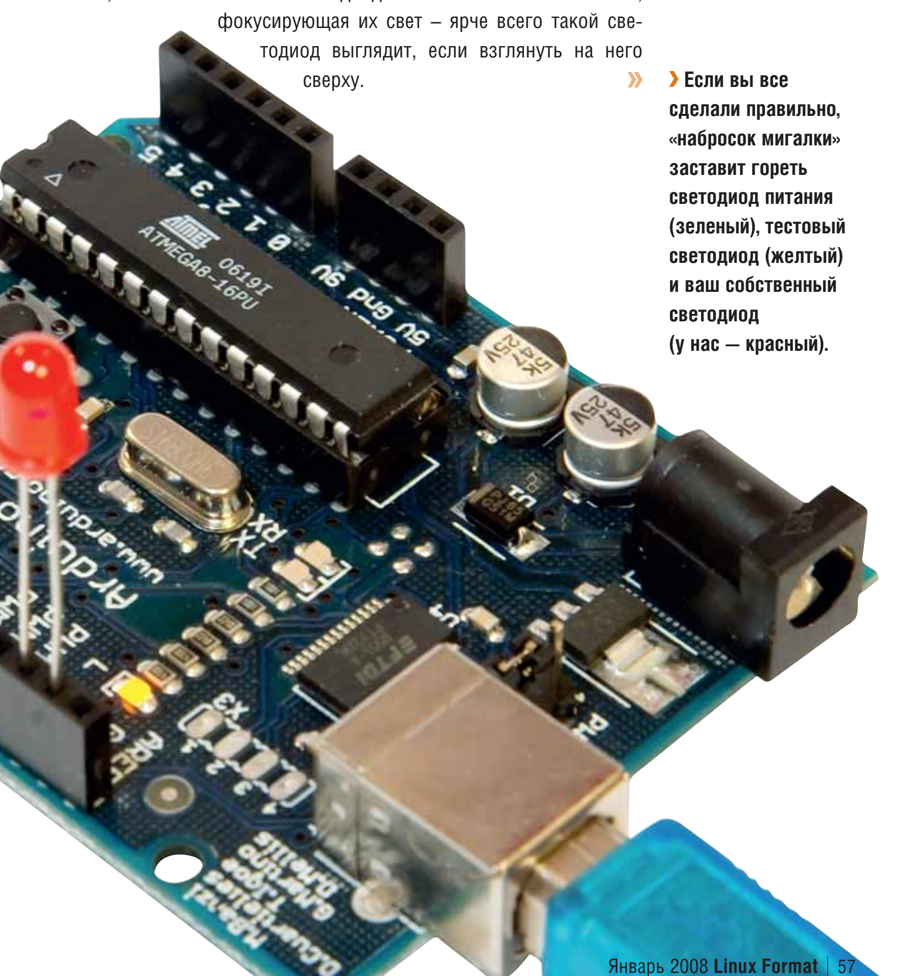
» **delay(300)** заставляет процессор сделать паузу в 300 миллисекунд, т..е. примерно на треть секунды.

» **digitalWrite(ledPin, LOW)** выключает светодиод.

» **delay(100)** заставляет процессор ждать одну десятую секунды.

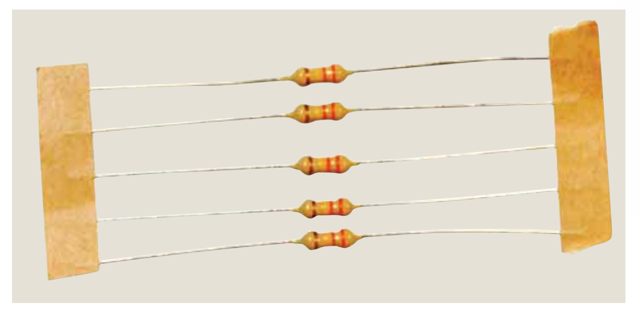
Вот и все! Функция loop() вызывается с частотой, равной частоте работы процессора, а вызовы delay() вставлены, чтоб разрешить процессору периодически перевести дух – иначе светодиод будет мигать так часто, что глазу не уследить, и покажется, что он всегда включен! Помните, что на большинство светодиодов насажена маленькая линза, фокусирующая их свет – ярче всего такой светодиод выглядит, если взглянуть на него сверху. »

› Соедините светодиод с цифровыми контактами 13/GND, да присмотритесь к длине его ножек, чтобы не перепутать полярность!

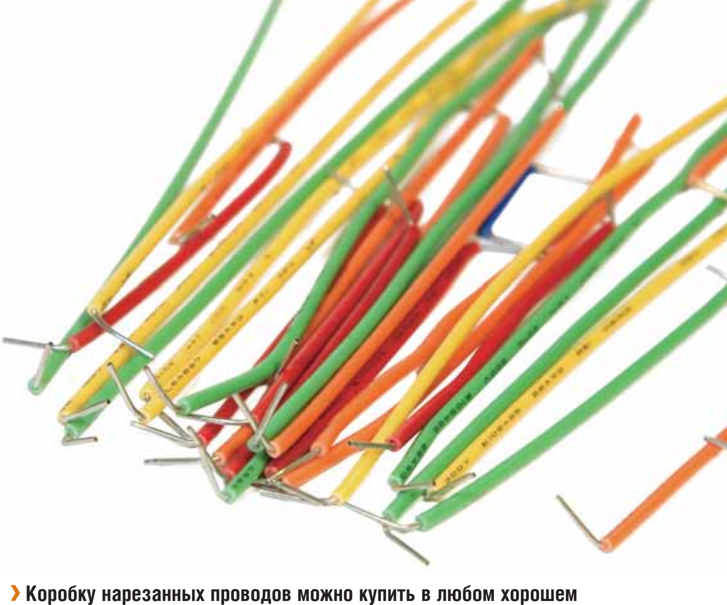


› Если вы все сделали правильно, «набросок мигалки» заставит гореть светодиод питания (зеленый), тестовый светодиод (желтый) и ваш собственный светодиод (у нас — красный).





› **Резисторы безумно дешевы и обычно продаются по пять штук; внимательно осмотрите полоски, чтобы определить их номинал.**



› **Коробку нарезанных проводов можно купить в любом хорошем магазине радиоэлектроники. Чтобы ваша схема смотрелась круто, используйте провода разных цветов для разных типов соединения!**

Использование макетной платы

Макетная плата имеет металлические дорожки, спрятанные под отверстиями и соединяющие ряды контактов в электрическую цепь. Колонки не соединены, поэтому горизонтальных контактов нет – есть только вертикальные. Если у вас полномерная макетная плата типа нашей, то на ней между рядов имеется просвет. Ряды по краям просвета тоже не соединены между собой. Рассмотрим на примере, как это работает: модифицируем простую светодиодную систему введением макетной платы и двух проводов. В моей плате колонки пронумерованы с 1 до 60, а ряды обозначены буквами от A до J, как показано на фото на стр. 56, и я воспользуюсь этими обозначениями для указания, куда втыкать провода. Если у вас таких обозначений нет, мои все равно вам помогут, подсказав, где ряд, а где колонка, а больше нам ничего и не надо.

> «Шпаргалка подскажет, какие полоски каким номиналам резистора соответствуют.»

Итак, для перенесения проекта «мигалки» на макетную плату, соедините проводом Gnd с J33, а другим проводом – Digital 13 с J32. Это подводит ток к контактам I33 и I32, H33 и H32, G33 и G32, F33 и F32. Моя плата имеет просвет между рядами F и E, поэтому E32 и E33 не входят в схему. Теперь вам осталось только поместить ваш светодиод на правильное место, чтобы замкнуть цепь; по-вашему, это куда?

Если вы не страдаете схемобоязнью, можете просто тыкать в контакты, пока светодиод не загорится – «методом тыка» вы много чего изучите, и это очень забавно, поверьте! В данном примере, светодиод загорится в позиции F33 (минус, короткая ножка) и F32 (плюс, длинная ножка); то же произойдет при помещении его в колонки G, H или I.

Попробуем сделать что-нибудь еще: с другой стороны платы Arduino находятся аналоговые входные контакты Analog In, а также несколько помеченных как Power. Ток в них будет побольше – заметили на одном из них отметку 5V, а на втором 9V? Это, между прочим, вольты; при таком напряжении нужна осторожность – подав на светодиод слишком много, вы его спалите. Если вы приволокли мешок светодиодов, беда невелика; но вдруг у вас их только три-четыре, и они последние! К слову, вполне безопасно подсоединить их на пару секунд, просто чтобы убедиться, что они работают. Пожалуйста, сделайте это сейчас: отключите Arduino от питания, затем соедините 5V с J33, а Gnd рядом с 5V – с J32. Потом на секунду включите плату: ваш светодиод ярко засветится (а заодно и нагреется!). Убедившись, что он работает, отключите USB-кабель, чтобы светодиод погас.

Перейдем к более сложной задаче: я хочу познакомить вас с резистором. Сия маленькая деталь задает падение напряжения, в зависимости от своего номинала сопротивления. Номинал изображен маленькими цветными полосками на корпусе резистора, но простым смертным тут ловить нечего – лучше держать под рукой шпаргалку, разъясняющую, что какие полоски означают. Теперь соедините 5V с J38 – так, чтобы светодиод не был подключен к питанию. Для замыкания цепи соедините резистором I38 и J33, и вы снова увидите свечение светодиода – но чуть тусклее. Если цветные коды резисторов для вас китайская грамота, попробуйте поочередно вставлять разные резисторы и рассортируйте их по яркости свечения светодиода!

Чтение ввода

В фирменном комплекте деталей имеется фотоэлемент – датчик, возвращающий различные значения в зависимости от падающего на него света. Потребуется чуть больше проводов, дополнительный вызов функции, а также условный оператор, но на самом деле не так все сложно – и стоит затраченных усилий: ведь вы напишете код, благодаря которому Arduino сможет ориентироваться в своей среде! Моя макетная плата снабжена двумя специальными дорожками, помещенными ради удобства подвода питания, но я ими пользоваться не буду, потому что платы поменьше не всегда обладают подобной роскошью.

Сперва поместите ваш фотоэлемент на макетную плату. Я воткнул его в F34 и F37. Поместите мощный резистор в G34 и соедините его с I29, затем соедините проводом J29 и 5V – это подаст питание на фотоэлемент. Необходимо также соединить проводом G37 и контакт Gnd рядом с 5V, для замыкания электрической цепи. Чтобы считать данные с фотоэлемента, соедините другим проводом H34 с Analog 0, и ваша система готова для программирования!

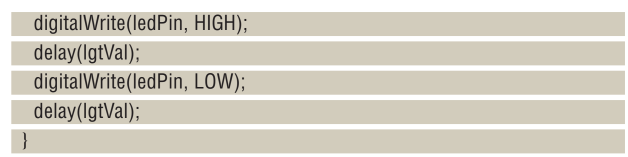
«Набросок» для чтения данных с аналогового входа в Arduino очень прост: вам необходимо указать, с какого контакта вы будете брать данные фотоэлемента, а также сохранить эти данные. Тут пригодится функция analogRead(), возвращающая значение, которое вы можете использовать для работы. В виде «наброска» это выглядит так:

```
int ledPin = 13;
int lgtPin = 0;
int lgtVal = 0;
void setup()
{
  pinMode(ledPin, OUTPUT);
}
void loop()
{
  lgtVal = analogRead(lgtPin);
```

Оборудование в LXF

Мы в *Linix Format* очень редко занимаемся оборудованием вроде Arduino – фактически, это наш первый опыт! Поэтому, пожалуйста, напишите, как вы думаете: нужно ли продолжать эту статью, с применением большего количества деталей, более сложных «набросков» и, возможно, даже каких-нибудь полных проектов? Или лучше держаться от «железок» подальше? Дайте нам знать! letters@linuxformat.ru

> «Чтение с аналогового входа в «наброске» Arduino очень простое — через функцию analogRead().»



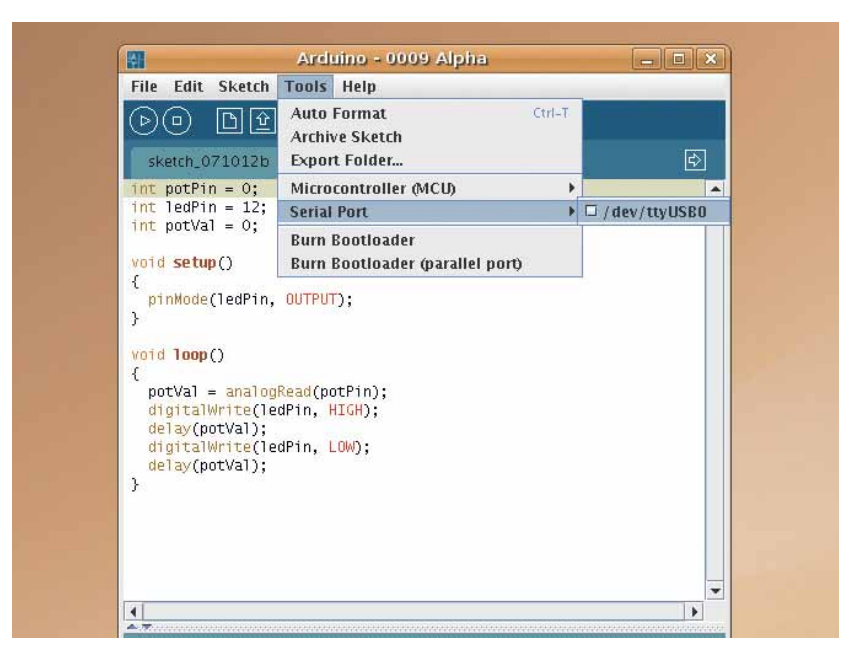
```
  digitalWrite(ledPin, HIGH);
  delay(lgtVal);
  digitalWrite(ledPin, LOW);
  delay(lgtVal);
}
```

Как вы думаете, что этот «набросок» должен делать? Если вы не уверены, попробуйте разобраться, загрузив его в вашу плату и прикрывая/открывая фотоэлемент!

Дерзать дальше...

Если вы раздобыли потенциометр, то заставить ваш фотоэлемент работать будет особенно легко: ваш провод от Analog 0 необходимо соединить со средней ножкой потенциометра; провод от Ground – с левой ножкой; а провод к питанию (через резистор) – с правой ножкой. Убедитесь, что потенциометр плотно вставлен в макетную плату – ножки у них обычно более толстые, так что придется на него поднажать!

Конечно, это только беглый обзор возможностей Arduino, и все потому, что настоящая магия заключается в программах, которые вы напишете. Мы показали вам, как собрать схему для чтения и записи данных, и следующий вопрос такой: что вы будете с ней делать? LXF



› **Перед попыткой загрузки в плату убедитесь, что для Arduino выбран правильный порт — это обычно /dev/ttyUSB0 или нечто подобное.**

Trinity

Станция по работе с цифровым звуком на базе открытого ПО может перевернуть рынок звукозаписи, которому недавно грозила монополия разорительно-дорогого и беспросветно проприетарного оборудования, как мобильного, так и студийного…



Наш эксперт

Дэниэл Джеймс осуществляет заказную разработку мультимедиа-продуктов для OEM-поставщиков; он – директор 64 Studio Ltd, что выпускает 64-битный дистрибутив GNU/Linux специально для творческой работы.

С тех пор, как викторианские изобретатели впервые нанесли музыку на цилиндры и диски, для музыкальной записи и воспроизведения применялось много разных технологий. Механические «резаки» и проволочные рекордеры сменились магнитной лентой как основным носителем, когда американский агент Джек Маллин [Jack Mullin] увел пару магнитофонов из нацистской радиостанции в конце Второй мировой войны. Аналоговые записи на пленке царили в музыкальной индустрии несколько десятков лет, а размеры лент варьировались от двухдюймовых студийных гигантов до крошечных Nagra в одну восьмую дюйма, без которых не обходился ни один шпионский фильм тех лет.

В 1980-х и 1990-х популярность обрели цифровые магнитофонные форматы DASH, DAT и ADAT, а затем их вытеснили портативные устройства на жестких дисках и флэш-картах. Основным преимуществом таких устройств являлась запись в файл, сделавшая возможным компьютерное редактирование и обработку. Для перевода в цифровой формат ленту пропускали через станцию по работе с цифровым звуком (digital audio workstation, DAW), на что уходили часы.

Однако тогдашние «безленточные» устройства предоставляли самые скудные возможности редактирования и крохотные дисплеи, не имея связи с интернетом. Правда, они были очень компактны, но аудиоматериалы для окончательной обработки все равно приходилось физически передавать на компьютер. DAW-звено этой цепи изначально было «приковано» к офису, так как аппаратные требования к аудиоредактированию в реальном времени превышали способности тогдашних ПК или ноутбуков. Для работы необходимы были специализированные проприетарные DSP-карты ценой в тысячи долларов.

С ростом скоростей процессоров в начале текущего десятилетия программы DAW получили возможность работать на центральных процессорах, без помощи звуковых карт. ПО DAW появилось и на ноутбуках, вне студий – звук записывается и редактируется прямо на месте. Главный недостаток такого подхода состоит в том, что аудиоаппаратура ноутбуков далека от профессиональных стандартов. То же относится к наладонникам типа Compaq/HP iPAQ и другим КПК – встроенная аппаратура годится для голосовых заметок, но никак не для качественной записи музыки.

Как выход из положения, инженеры мобильной звукозаписи применяют внешние аудиоинтерфейсы, обычно с подключением через FireWire, для получения сносного входного качества. Этот комплект вполне можно утащить на себе, но размер его отнюдь не карманный: на столе он раскладывается без проблем, а в дорожную сумку уже не влезет. Портативности шпионских Nagra прошлого века нынешняя аппаратура звукозаписи пока не достигла.

Микшер в полете

Рональд Стюарт – режиссер звукозаписи из Сиэтла, ныне живет в Лос-Анджелесе. Он был среди первых пользователей системы ProTools от Digidesign, де-факто стандарта проприетарных DAW с начала 1990-х. В то время Стюарт записывал ремиксы для убыточного какофонистского поп-дуэта The KLF и ему подобных, а через пару лет задумался о создании DAW «под себя»: портативной, простой в использовании и способной долго работать без подзарядки – а главное, обеспечивать первоклассное качество аудио без лишних ящиков и кабелей.

В идеологии «гибкой» (Agile) разработки ПО каждый проект начинается с истории пользователя. История Стюарта была в том, что он – звукорежиссер, проводящий много времени в разъездах. Компьютер его мечты должен был позволять по своим габаритам комфортно устроиться с ним в самолетном кресле, а также обеспечивать ресурсоемкую обработку аудиоданных на время всего рейса без подзарядки. Кроме того, требовалась возможность делать аудиозаметки и редактировать звукозаписи, а по прибытии в аэропорт выгружать результаты в студию через Wi-Fi.

Устройство-прототип

Сперва Стюарт заинтересовался игровой консолью Sony PlayStation Portable. Он рассудил, что такое устройство, гораздо меньше ноутбу-

Будущее проекта

Первый прототип проекта Trinity, состоявший из специальной платы и прочих частей, помещенных между двумя слоями оргстекла, был впервые представлен публике на 5-й Международной конференции по Linux-аудио в Берлине, в марте 2007. Второй прототип, в более практичном алюминиевом корпусе, дебютировал на выставке Podcast and New Media Expo в Онтарио, Калифорния, шестью месяцами позже. Короткая линейка из семи устройств Ontario отобразила стремление конструкторов уменьшить и облегчить прибор; этот подход будет воплощен в окончательной версии продукта, выпуск которого намечен на 2008 год. Ожидаются также существенный подъем в производительности процессора, пропускной способности диска и количестве памяти, ведь темп прогресса на ультра-мобильном фронте стремителен.

Проект дистрибутива Transmission организован по образу и подобию любой инициативы Open Source, с собственным базовым сайтом и списком рассылки. Подключение к проекту приветствуется, а так как дистрибутив x86-совместимый, то специальное оборудование для участия не требуется. Любой 32- или 64-битный x86-совместимый компьютер сойдет за платформу разработки, только вот двойная загрузка в проекте не предусмотрена, поэтому желательна отдельная машина. Дистрибутив основан на Debian, и его обновление несложно выполнять с помощью скрипта *trinit-upgrade* (это оболочка для различных APT-вызовов). Мы ждем ваших предложений по поводу Trinity, и надеемся, что этот проект станет не только первым мобильным аудио-компьютером с GNU/Linux и свободным ПО внутри, но и признанной платформой для музыкантов, звукорежиссеров и репортеров всего мира.



› Берлинский прототип со снятой верхней частью корпуса.



› Второй прототип. Еще не портативный, но уже работает.

ка по размеру, но с достаточно мощным процессором, вполне может нести на себе профессиональное аудио-ПО, при достаточно качественном аудио-интерфейсе. Так как PSP официально закрыта для сторонних разработчиков, Стюарт собрал прототип, использовав маленькую системную плату VIA x86 и 6,5-дюймовый TFT-дисплей.

Опыта применения свободного ПО в студии у него не было, но он слышал, что платформа GNU/Linux достаточно эффективна в работе со встроенными устройствами с низким энергопотреблением, и обладает набором мультимедиа-приложений. Несколько производителей профессионального аудиосектора уже выпустили встроенные продукты на основе ядра Linux, сократив затраты на разработку и ускорив выход на рынок благодаря изобилию готовых блоков свободного кода. Расчет был также и на независимость от разработчика ОС.

В первом прототипе Стюарта, Xubuntu (вариант Ubuntu с десктопом Xfce) был установлен на миниатюрный жесткий диск в стиле iPod. С помощью *Audacity* через аудио-интерфейс Echo Audio было сделано несколько пробных записей, результаты которых убедили Стюарта в том, что он на верном пути. Он назвал проект Trinity и организовал группу под названием Trinity Audio Group для коммерческой разработки устройства.

Для второго прототипа у Compulab были заказаны модуль CM-iGLX и макетная плата. CM-iGLX, размером с кредитную карту, оснащен мобильным процессором AMD Geode LX800 с тактовой частотой 500 МГц, 256 МБ DDR ОЗУ и 512-МБ флэш-накопителем, и по составу интерфейсов почти не уступает «настоящим» материнским платам. Процессоры Geode LX используются в проекте One Laptop Per Child; они x86-совместимы и могут выполнять стандартный дистрибутив, а оптимизация для таких процессоров недавно была включена в ядро Linux.

CM-iGLX обладает потребительскими аудиовозможностями, включая кодек AC97, но требования к качеству звука в проекте Trinity вынудили разработчиков заказать специальную плату. К тому же в большинстве студий применяются электростатические микрофоны, с предусилителями для питания цепи, сходными со схемой питания шин USB отсутствием внешнего источника. Потребительские аудиовозможности такого не предусматривают, и это стало второй причиной, побудившей Стюарта заказать специальную плату. В нее был встроен лицензированный у Echo Audio аудиочипсет высшего качества. Модуль CM-iGLX вставляется в гнездо на материнской плате, оснащенной также Neutrik-комбинацией микрофонного/линейного ввода с регулятором уровня, гнездами линейного вывода, выводом для наушников с регулятором громкости, стандартными USB-портами и гнездами питания.

Специализированный дистрибутив

Я встретился с Рональдом Стюартом в Лос-Анджелесе в декабре 2006, узнав о проекте Trinity из пресс-релиза на Linuxdevices.com: Стюарт показал мне второй прототип на основе модуля CM-iGLX и макетной платы, работающий под управлением Xubuntu. Моя компания недавно выпустила дистрибутив 64 Studio 1.0, и я сразу понял, что мы могли бы значительно улучшить ПО устройства. Я ничуть не умаляю достоинств Xubuntu и Ubuntu как дистрибутивов общего назначения, но при профессиональном создании и редактировании музыки неизбежен нестандартный подход. Например, Trinity по умолчанию не оснащается QWERTY-клавиатурой, а дисплей очень мал по меркам ПК или ноутбуков – всего 8 дюймов по диагонали, с разрешением 800x600.

> «У некоторых изготовителей профессиональной аппаратуры уже есть продукты на базе Linux».

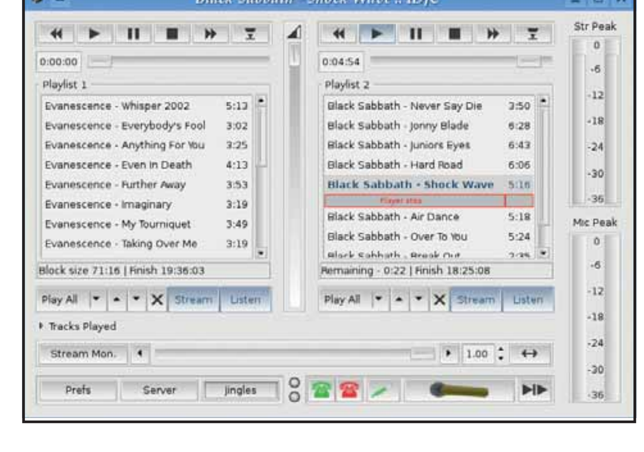
Кроме того, упор на качество звука и недопустимость сбоев требуют наличия ядра реального времени, которое по умолчанию в обычные дистрибутивы не входит. В 64 Studio мы накопили солидный опыт по включению RT-патчей Инго Молнара [Ingo Molnar] в несколько последовательных версий Linux-ядра, а также по использованию тех- »

» нологий типа rtlimits, обеспечивающих безусловный приоритет аудио-процессов. Интерфейс может зависнуть, запись на диск подождет, но на концерте «живой» музыки в присутствии 10 000 зрителей звук не имеет права прерываться при увеличении системной нагрузки. То же справедливо и для важных живых аудиозаписей, когда повторить звук дважды невозможно.

Мы предложили Стюарту создать специальный дистрибутив для Trinity. Как основа, подошел бы Debian 'Etch', с рядом модификаций, сделанных нами для дистрибутива 64 Studio и одного OEM-заказчика. Система Custom Debian Distribution, сделанная нашим ведущим разработчиком Фри Эканаяка [Free Ekanayaka], позволяет создавать специализированные Debian-дистрибутивы при высокой степени автоматизации многих рутинных задач и максимальной унификации основных пакетов.

Кроме того, имея в виду привычные для аудиоинженеров тумблеры, мы посоветовали предусмотреть особый пользовательский интерфейс, под который приспособили хорошо знакомый нам *Fluxbox*: он стабилен и чрезвычайно неприхотлив к системным ресурсам. Нам хотелось сберечь каждый такт процессора для обработки аудио: хотя Geode поразительно скор для встроенного процессора с потреблением энергии всего несколько ватт, но одноядерный чип x86 на 500 МГц вряд ли потянет современный рабочий стол, и надо было использовать его эффективно. Анимированные курсоры и 3D-эффекты в подобном случае – пустой перевод ресурсов: не говоря о нагрузке на процессор, к тому же часто отсутствует аппаратное ускорение графики.

Специализированный дистрибутив для Trinity был назван Transmission. Программа установки взята у Debian, но все вопросы уже разрешены, и пользователю незачем знать о существовании Debian или GNU/Linux при установке дистрибутива на чистый или отформатированный диск. Программа установки англоязычная, настроена на US-раскладку клавиатуры (портативная QWERTY-клавиатура подключается к USB) и один из часовых поясов США. Пользователь единственный, **trinity**, вход без пароля. Пакет *gdm* включен, и можно добавить несколько учетных записей с паролями. Использовать виртуальную клавиатуру при входе не получится, потому что она активируется уже после входа. Мы планируем авторизацию через USB-ключи для более безопасного многопользовательского входа без клавиатуры.



› *Internet DJ Console* оснащена всем необходимым для работы.

Выбор приложений

После загрузки *Fluxbox* появляется девять значков, по числу основных приложений. Они настраиваются инструментом *idesk* и должны постоянно находиться в четырех рабочих местах, для быстрого запуска приложений без использования меню. В последних версиях интерфейса пакет *fbpanel* заменил стандартную панель задач *Fluxbox* рядом мелких значков, предназначенных для запуска вспомогательных приложений, типа *xvkbd* (экранная QWERTY-клавиатура для ввода названий файлов в аудио-приложениях).

Сердце системы – пакет *jackd*, Jack Audio Connection Kit (комплект аудио-подключений, известный большинству пользователей как просто 'Jack'). Jack – очень эффективная виртуальная коммутационная панель, направляющая аудио от одного порта к другому, и в сочетании с ядром реального времени способен обеспечить бесперебойную работу. Эта программа также управляет синхронизацией различных аудио-приложений и перемещением в пределах аудиозаписей (аналог кнопок воспроизведения, паузы и перемотки назад и вперед в обычном магнитофоне). На новейших процессорах Jack может свести время запаздывания аудио в миллисекундах к считанным единицам, неразличимым для человеческого уха. Jack настроен так, чтобы время задержки было не хуже, чем если бы единственное приложение имело монопольный доступ к аудио-интерфейсу.

Потребности Trinity в записи, редактировании и микшировании обеспечивает *Ardour* – DAW-приложение на основе Jack, исходно созданное для крупных систем с несколькими мониторами с разрешением как минимум 1280x1024. У *Audacity* интерфейс куда проще и скромнее, но поддержка Jack в нем осуществляется через библиотеку *PortAudio*, и она является менее «родной» по отношению к Jack, чем *Ardour*. *Ardour* также обладает преимуществом DSP-эффектов реального времени, предоставляемых модулями расширения в формате LADSPA. Зная, что *Ardour* предстоит работа с диском сравнительно малой пропускной способности, из-за миниатюрного и медленного дисковода, Эканаяка так модифицировал интерфейс *GTK+*, чтобы он выводил на экран лишь наиболее часто используемые органы управления. В результате *Ardourino*, как мы назвали новую версию, уместился на дисплее 800x600.

› Задняя панель Ontario (видны вводы/выводы аудио и блок USB Wi-Fi).



› *LamSynth* – свободный эмулятор классического синтезатора с верньерной настройкой.



› А вот *Ardourino*, запущенный на прототипе Ontario. Не путайте с Arduino!

Ссылки

- Домашняя страница Trinity www.trinityaudiogroup.com
- Сайт разработки Transmission http://trac.64studio.com/trinity
- Ardour www.ardour.org
- amSynth http://amsynthe.sourceforge.net/amSynth/
- Ударная установка Hydrogen www.hydrogen-music.org
- Internet DJ Console www.onlymeok.nildram.co.uk

Виртуальная клавиатура

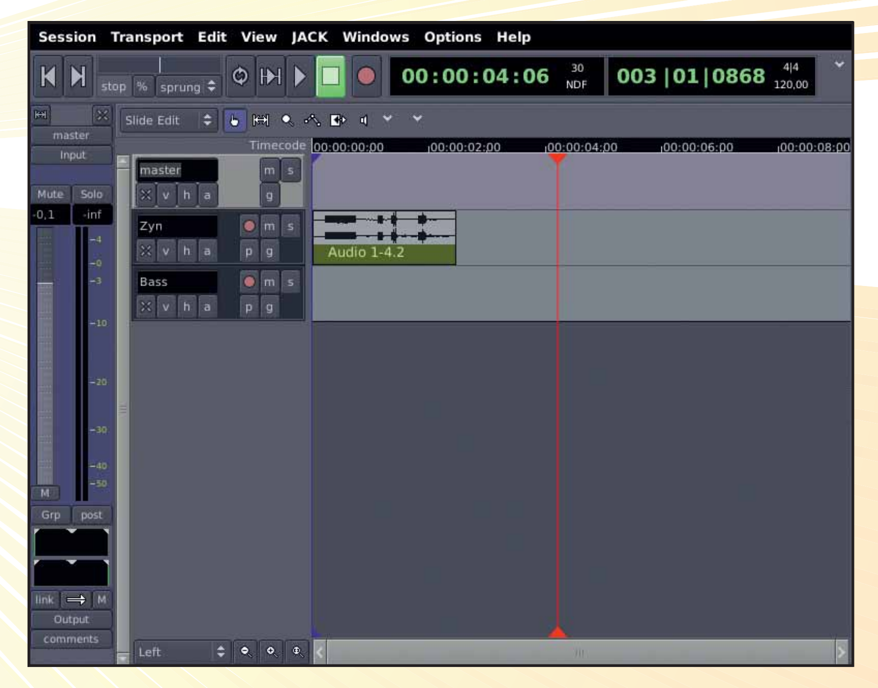
В дистрибутив включено приложение *amSynth*, программная модель аналогового двухосцилляторного синтезатора, и виртуальная MIDI-клавиатура, на которой можно играть курсором мыши. Включен и секвенсор ударных *Hydrogen*, с набором синтезированных барабанов и сэмплов для них. *amSynth* может задействовать внешние MIDI-устройства, подключаемые через USB. И *Hydrogen*, и *amSynth* способны реагировать на сигналы подключаемых MIDI-устройств, например, портативной USB-клавиатуры на батареях. Поэтому Trinity – прибор, задуманный как инструмент для записи и редактирования – вполне пригоден как инструмент для создания музыки и/или консоль для сочинения и микширования «живых» мелодий.

Одно из приложений, активно эксплуатирующих беспроводные и Ethernet-подключения Trinity – это *idjc*, Internet DJ Console. На первый взгляд, это обычный медиа-плейер для ди-джея, с двумя списками воспроизведения и кросс-фейдером, но это также и клиент потокового радио Icecast/Shoutcast. У него есть и другие свойства, полезные в среде радиовещания: кнопки наигрышей, микрофонное сжатие и даже интеграция интернет-телефонии VoIP, так что с помощью *idjc* можно вести собственное телефонное шоу: подключить микрофон и вещать в «живом эфире» из любой точки земного шара (при наличии доступа к Wi-Fi) – не будучи привязанным к месту паутиной кабелей.

В Transmission включены и другие приложения, используемые в процессе обработки звука. Например, *Gnome CD Master* может прожигать аудиофайлы на CD, при подключении через USB внешнего пишущего CD-привода. Это приложение не очень известно в среде Open Source, но оно обладает некоторыми уникальными функциями – например, помещением маркеров дорожек непосредственно в аудио-сигнал «на слух», очень полезным при разбивке продолжительных записей. При работе с Сетью для выгрузки файлов используются *gFTP* и легкий браузер *Epiphany*, а удаленный вход осуществляет *openssh-server*. Проводные и беспроводные DHCP-подключения контролируются пакетами *wifi-radar*, *wireless-tools* и *network-manager*. LXF

› *Ardour* – неплохая DAW, но создана для больших дисплеев.

› Поработав с *GTK+*, интерфейс удалось втиснуть в гораздо меньший экран.



Мнение: Джереми Эллисон

Извела меня кручина...



Джереми Эллисон

Джереми – ведущий разработчик команды Samba, в настоящее время работает в Google, в Сан Хосе, Калифорния.

Для моего кода это был не самый удачный месяц. Проекту Samba, за который я отвечаю, пришлось объявить о нескольких уязвимостях. К несчастью, некоторые из них содержатся в коде, который писал я сам. Когда такое случается, я всегда копаюсь в своей душе. Нет ничего хуже, чем узнать, что тысячи людей тратят время на поиски заплаток, а причина — твой проект. В подобных случаях я задумываюсь: не пора ли завязать с программированием и посвятить остаток дней мирной стезе менеджера, исследуя коды других вместо создания собственного?

Проект Samba не юн: первичный код создавался пятнадцать лет назад. Тогда мы и не слыхивали о таких современных проблемах безопасности, как переполнение целого (сумма двух целых чисел может оказаться меньше, чем любое из слагаемых, из-за выхода за разрядную сетку процессора) или кучи (перезапись невыделенной памяти, позволяющая злоумышленнику перехватить управление программой). Мы знали о переполнении буфера (запись в буфер количества данных, превышающего его емкость) и о DDoS-атаках, но 1992 год был проще и дружелюбнее к разработке сетевых программ. Большинство установок Samba проводилось в сетях, изолированных от Интернета, технически грамотными администраторами.

Ошибки в системе безопасности, обнаруживаемые сегодня, намного сложнее и изощренней. И относимся мы к ним намного серьезнее. В те ранние дни мы работали с людьми, знающими, как компилировать исходный код. Ответная реакция на «дыры» наступала через считанные часы. Первая попытка саботировать Samba через эксплойт нулевого дня была направлена в наш список рассылки без всякого предупреждения. Эндрю Триджелл [Andrew Tridgell] (автор первой Samba) пришел в ярость и сработал на рывок, чтобы уже следующее сообщение в списке было заплаткой для этой проблемы. И это ему удалось. Тогда мы редко получали эксплойты нулевого дня, открыто отправленные в списки рассылки. Плохие парни держат их в секрете для своих собственных противозаконных целей, а хорошие парни практикуют наилучший подход, принятый в наши дни – отправка уведомлений непосредственно разработчикам Samba (security@samba.org) и сотрудничество с нами по координации уведомлений о проблеме и поиска путей ее решения.

Ныне ответная реакция значительно медленнее, даже при том, что на нашей внутренней кухне мы обычно находим решение за несколько часов; тормозит нас свой же успех. Чтобы с полной ответственностью выпустить заплатку, мы согласуем наши действия с огромным количеством поставщиков, которые продают и перепродают наш код. У большинства из них есть свои графики выхода заплаток и команды по безопасности, которые оценивают предложенные решения и согласовывают, когда именно объявить об уязвимостях по всему миру.

Кто обнаруживает наши ошибки? Сейчас наблюдается соотношение пятьдесят на пятьдесят между внутренним аудитом и специалистами по безопасности из мира Open Source и внешними специалистами по безопасности, которые зарабатывают на жизнь, получая контракты благодаря своему умению отыскивать уязвимости в открытых проектах. Заметив «дыру» в безопасности, мы сразу же проверяем всю кодовую базу на предмет похожих проблем или аналогичных практик написания кода, способных в будущем стать причиной проблем, и переписываем или удаляем их. Так получается дольше, но зато намного надежнее в конечном итоге. Взглянув на код Samba, вы найдете, что стандартные функции, известные своей ненадежностью, просто не поддадутся компиляции в случае добавления к нашему коду. Набор автоматических макросов предупреждает о любом использовании известных «вредоносных» функций. Мы не можем полагаться на автоматику по части защиты: все пока что существующие уязвимости нашей системы безопасности подразумевают использование некоего допущения в одной части кода, вызывающего появление уязвимости в другой, якобы отдельной части кода, которая на самом деле является взаимозависимой. И злоумышленники, и исследователи уязвимостей в наши дни очень поумнели.

Проще ли обнаруживать и бороться с ошибками в системе безопасности программ с открытым кодом, чем в проприетарных программах? По моему ощущению – да. Возможность изучить намерения автора, посмотрев его код и комментарии, очень помогает увидеть, где идея не соответствует коду. Там-то и скрываются проблемы безопасности. Надежнее ли открытый исходный код, чем проприетарный код? И на это я бы тоже ответил «да». Одно лишь количество выявленных уязвимостей ни о чем не говорит: примерно половина сбоев в системе безопасности обнаруживается в ходе наших внутренних проверок кода, или просто кто-то просматривает код и думает: «ага, вот оно…» Что бы мы ни обнаружили, мы обязаны сообщать об этом нашим контрагентам и оповещать о сбое в системе безопасности. Это штука болезненная, но поскольку все наши процессы открыты, мы не можем позволить себе роскошь тихонечко, молчком, исправить ошибку в надежде, что никто ничего не заметит. А вот проприетарные производители могут себе эту роскошь позволить, и не пользоваться ею было бы безумием, и я уверен, что они и пользуются. Проприетарные производители цепляются за любую возможность, конкурируя с замечательным свободным ПО.

ПроСИм, проСИм

Samba все еще пишется на C. Мы рассматривали возможность перехода на C++, но это не поможет решить наши проблемы с безопасностью. На самом деле, если скрыть взаимозависимости в классах объектов, то отслеживать эти ошибки, возможно, станет еще труднее. Перевод кода на управляемый язык, например, Java или C#.NET, мог бы сократить наши проблемы с безопасностью, но заодно снизил бы производительность – а именно она для многих является решающим фактором в выборе Samba. Не случайно ни один из конкурирующих серверов CIFS не написан в управляемом коде. У нас ушло бы от шести месяцев до года на то, чтобы переписать все и выйти на тот же самый уровень функциональности, вероятно, снизив производительность. Но у нас для этого нет ни времени, ни людей. Если кто-то из вас занимается созданием хорошего свободного сервера CIFS, я бы предложил вам создать его на управляемом языке – в качестве эксперимента. Если он станет работать так хорошо, как я надеюсь, мы выбросим свой код и присоединимся к вам.

Чтобы Samba шла в ногу с современными требованиями, мы постоянно переписываем и переделываем основные части кода. Наш грядущий релиз, Samba 3.2, имеет очень мало общего с оригиналом 1992 года и использует современные технологии управления памятью, дабы избежать переполнения кучи, буфера и утечки памяти, причем большинство этих проблем были обнаружены командой Samba при внутреннем пользовании. Мы сами используем Samba для обслуживания файлов. Зализывая свои раны, мы врачуем и других.

Я по-прежнему чувствую себя скверно из-за допущенных ошибок, но пробую утешиться, предлагая их как учебный пример при приеме на работу в Google. Если кто-то не видит ошибки, это всегда поднимает мне настроение, но в то же время служит большим минусом для претендента. Так или иначе, новые сотрудники должны уметь писать код лучше меня… (!) Код открыт для изучения, так что учтите мои ошибки, чтоб не пришлось согласовывать время объявления о «дырах». Такого познавательного опыта лучше не иметь. LXF

Учебники

Наши эксперты помогут вам с любым приложением Linux



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Инструмент для специалистов

Цивилизация в подлинном смысле слова состоит не в умножении потребностей, а в свободном и хорошо продуманном ограничении своих желаний.

Мохандас Карамчанд (Махатма) Ганди

GNU/Linux очень давно используется как инструмент самыми разными специалистами. Научное сообщество применяет его везде, где нужна автоматизация или присутствует уникальное оборудование. Программистское – везде, где нужен недорогой, надежный и гибкий конструктор. Потихоньку подтягиваются дизайнеры и «игростроевцы». Медленно, но верно выстраивается удобная среда для юристов и «продажников».

Операционная система и ее окружение – это чрезвычайно интересные объекты, но подавляющее большинство пользователей оценивают их полезность исключительно с точки зрения своей специализации: они должны выполнять какую-то работу. С точки зрения ученого – собирать и представлять данные. С точки зрения дизайнера – создавать и тиражировать образы. С точки зрения юриста – находить и генерировать информацию.

Для достижения своих целей специалисты подчас отходят от проторенных дорожек – им приходится делать нестандартные вещи. В этом они вполне могут пожертвовать лишней функциональностью ради возможности доработки инструментов. Не беда, что *Maxima* уступает по охвату той же великолепной *Mathematica*, зато чуть поправленную *Maxima* можно разместить в личном коммуникаторе. Не беда, что OCR *CuneiForm* пока что уступает популярному *FineReader*, зато отпущенные будущей весной на свободу исходные тексты программы можно будет использовать для обучения роботов чтению. Собрать читающего робота – это задача для специалиста. Это действительно интересно заняться и сделать это можно, только если под рукой есть поддающийся усовершенствованию универсальный инструмент. GNU/Linux вполне подходит для этих целей.

E.M.Baldin@inp.nsk.su

В этом выпуске...

Совет месяца: Потерянные в пространстве



Обращали ли вы внимание на странный факт: чем больше у вас жесткий диск, тем быстрее он заполняется? Всего каких-то пару месяцев назад эти сотни гигабайт казались прорвой свободного места, заполнить которое не под силу ни одному человеку – и вот его больше нет. Избыток дискового пространства делает людей ленивыми и, рано или поздно, они жалеют об этом. Но как узнать, куда девалось свободное место? Первое что приходит на ум – команда du (Disk Usage):

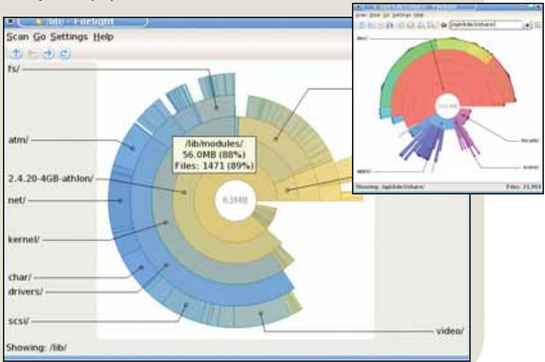
```
du -sch *
```

Она показывает суммарный объем каждого элемента текущего каталога. Удобно отсортировать этот список, и здесь du приходят на помощь другие утилиты Unix:

```
du -sm * | sort -n | tail -n 10
```

du -sm выводит размер каждой записи в мегабайтах, sort -n сортирует их, как числа (без -n, 440 предшествовало бы 45) а команда tail выводит десятку самых жадных каталогов. Найдя директорию, занимающую слишком много места, спускайтесь на уровень ниже – пока, наконец, виновник не будет установлен.

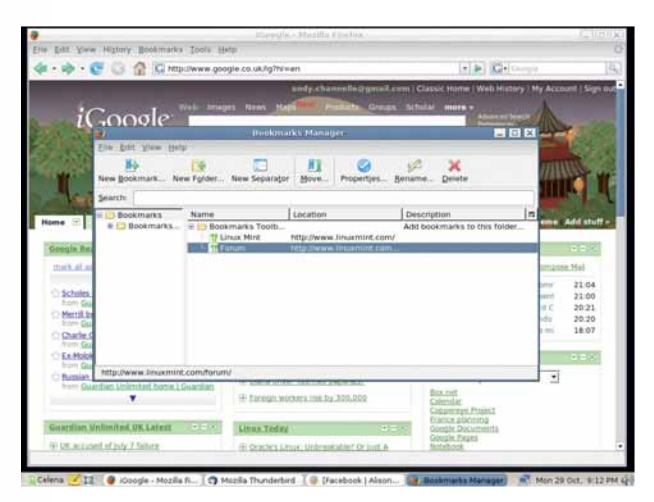
Конечно, не все любят командную строку. В качестве альтернативы можно попробовать *Filelight* (www.methylblue.com/filelight) – графическое приложение, изображающая каждый каталог в виде круговой диаграммы. Сканирование крупного домашнего каталога потребует времени, зато вы получите весьма и весьма ценную информацию.



Не застревай

Переход с Windows на Linux кое-кого пугает, но сейчас дистрибутивы достигли такой зрелости, что весь процесс прост как никогда, утверждает **Энди Ченнел**.





› Свежеустановленный Linux на вид пустоват, но чуть-чуть поработаем – и все будет иначе!



Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется со времен Dragon 32.

Переход с Windows на Linux раньше был травмоопасным мероприятием; данные могли быть в несовместимом формате, что-нибудь терялось, и в итоге вам оставалось только недоумевать: а стоила ли овчинка выделки? Однако те времена прошли: поставщики дистрибутивов и разработчики ПО серьезно поработали над совместимостью большинства вещей и над сетевыми технологиями.

На данном уроке мы обсудим наилучшие форматы для переноса документов и файлов из мира Windows на разделы Linux; а также как перейти на *Thunderbird* без потери писем и как сделать вашу web-историю совместимой, чтобы не пришлось заново откапывать те сайты, которые вы уже добавляли в закладки *Firefox*.

В последней версии Ubuntu, Gusty Gibbon, есть замечательный инструмент для переноса ваших Windows-настроек в Linux. Он называется Ubuntu Windows Migration Assistant (WMA) и создан для извлечения Избранного из *Internet Explorer*, закладок из *Firefox*, списков контактов из программ обмена мгновенными сообщениями AIM и Yahoo, а если его попросить, так он даже перетащит обои рабочего стола Windows на рабочий стол Linux. Первые наши тесты показали, что работает приложение вполне хорошо, хотя ряд вещей оно делать не умеет. Но проблемы здесь нет, потому что с этими задачами отлично справятся другие свободные приложения.

Предположим, что вы уже установили Linux и еще не успели удалить раздел Windows. Это важно, потому что на нем могут находиться компоненты, необходимость которых вы осознаете только минут через пять после форматирования **C:**. Поэтому советуем сохранять Windows-раздел, даже если вы не намерены его использовать, вплоть до удачного завершения вашего перехода. Рассматривайте его как точку восстановления системы или как резервную копию.

Форматы файлов

Если вы следите за новостями мира компьютеров, то, возможно, знаете, что сейчас бушует война файловых форматов. На одной стороне выступает Microsoft, которая пытается провести свой формат Office Open XML (OOXML) в качестве международного, чтобы затмить успех введенного в *OpenOffice.org* формата Open Document (ODF), который уже является «стандартным форматом» документов для правительств и разработчиков офисных приложений. Такие сообщения появляются постоянно, и поэтому важно поразмыслить над тем, как переносить документы из одной ОС в другую, и грамотно выбрать формат файлов.

Начнем с наиболее употребительного формата файлов наиболее часто используемого приложения: **.doc** файлы от *Microsoft Word*. Хорошая новость: **.doc** – довольно старый формат, и проект *OpenOffice.org* (*OOo*) сумел его разгадать. А значит, большинство документов Word будут открыты и нормально обработаны в *OOo*. Здесь могут возникнуть проблемы с макросами Visual Basic for Applications (VBA), поэтому если вы имеете дело с такими документами, то, возможно, захотите присмотреться к версии *OOo*, распространяемой компанией Novell в составе настольной операционной системы SUSE Linux (а также доступной для других дистрибутивов на сайте http://go-oo.org) и имеющей хорошую поддержку макросов VBA.

Еще лучший вариант, требующий либо немного дополнительного времени, либо много заблаговременного планирования (а то и машины времени): сохранить все свои документы в более открытых форматах, вроде RTF. Но опять же, он подходит только для несложных файлов: писем, книг, любовных записочек и других, содержащих в основном текст без макросов, сложных таблиц и вложенных данных.

» **Месяц назад** Мы выбирали облегченный дистрибутив для старого компьютера.

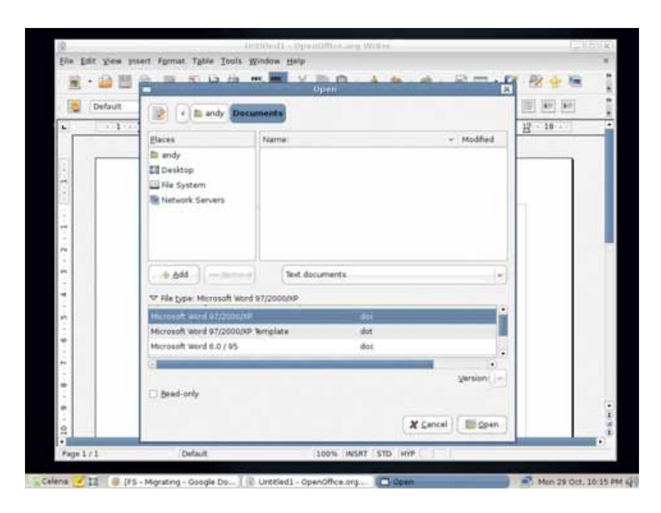
на переходе!

Если у вас много документов *Word* в разделе Windows, попробуйте перенести их в раздел Linux документ за документом. Например, вам нужен доступ к документу **story.doc**, находящемуся на разделе Windows. Откройте его как DOC-файл в *OpenOffice.org*, сразу же выберите в меню Файл пункт Сохранить как... и сохраните новый документ в папку **Documents** раздела Linux – либо с расширением **.odt**, если документ сложный, либо с расширением **.rtf**, если вы собираетесь послать документ другому пользователю или намереваетесь снова использовать его в *Word*. Эта операция отнимет всего несколько секунд вашего рабочего времени, причем только при первом открытии документа, а в будущем вы не попадете в переплет из-за непригодного к использованию **.doc**-файла. Теперь вы можете рассматривать исходный **.doc** как резервную копию, а всю текущую работу делать в новой сохраненной версии.

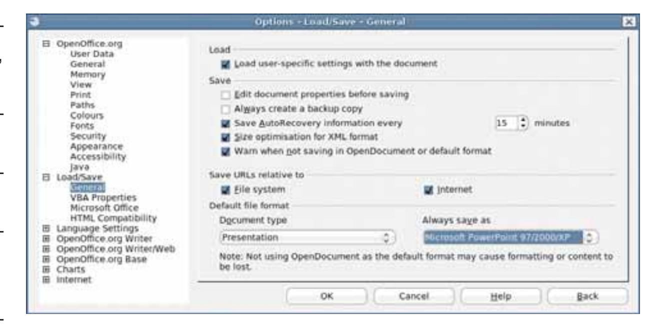
Другие офисные приложения для Linux тоже открывают и сохраняют **.doc** файлы, но *OOo*, по нашему опыту, надежнее всех прочих с точки зрения точности воспроизведения документов. В редких случаях, когда что-нибудь сработает неправильно, *OOo* сумеет исправить любую ошибку, и после сохранения файла с расширением **.odt**, все снова будет в порядке.

Если вы используете *OOo* в Windows и хотите перенести все настройки и шаблоны из Windows в Linux, то вы найдете все, что касается *OpenOffice.org*, в **C:\Documents and Settings\имя_пользователя\Application Data\OpenOffice.org2**. Последняя цифра указывает номер установленной у вас версии *OOo*. Скопируйте содержимое этой папки в скрытый каталог *OOo* вашего каталога **/home**, который называется как-нибудь наподобие **.ooo-2.0** или **.openoffice**. Когда вы после этого запустите приложение, все настройки должны быть правильными.

Другая альтернатива для несложных документов *Word* – это зайти на онлайн-сервис Docs от Google и загрузить все туда. Таким образом, вы «вынесете за скобки» все ваше локальное хранилище на удаленный узел и сможете менять ОС как перчатки без потери данных. Войдя на сервис под своей учетной записью, выберите на панели инструментов пункт Загрузить. Найдите нужный вам файл, введите имя документа, если оно отличается от имени файла, и нажимайте кнопку Загрузить файл. Через некоторое время (зависящее от вашего компьютера и скорости сети) вы увидите ваш документ внутри интерфейса Google Doc. Данный подход содержит больше ограничений по сравнению с переносом документов непосредственно в *OOo* (например, невозможность работы с колонтитулами и другими полезными вещами), но, с другой стороны, вам не придется беспокоиться о резервных копиях или доступу к определенному компьютеру с целью редактирования документов в будущем.



› *OpenOffice.org* откроет большинство документов *Microsoft Office*; избегайте самого последнего формата, .docx, и все будет в порядке.



› В *OOo* можно задать формат сохранения документов по умолчанию — например, .ppt.

Естественно, ни один из этих методов не поможет, если вы собирали ваш архив **.doc**-файлов 20 лет, но для среднего пользователя они вполне сгодятся.

Все это справедливо и для файлов в форматах **.xls** (*Microsoft Excel*) или **.ppt** (*PowerPoint*). *OOo* и другие приложения откроют файлы в форматах **.xls** и **.ppt**, но, как и в случае с *Word*, важно при любой возможности использовать стандартные, а не проприетарные форматы. Правда, с этими типами данных будет сложнее, поскольку для них нет хорошего эквивалента, как **.rtf** для файлов *Word*, за исключением форматов таблиц данных, разделенных запятыми (**.cvs**), или **html**, соответственно. Но и они подходят только для самых простых документов, так что лучший вариант – это открыть файлы в приложениях *OOo*, связанных с таблицами или презентациями, а затем сохранить их в стандартных форматах, как было рассказано выше.

Не сочтите за кощунство, но мы всегда использовали *OOo Impress*, а сохраняли в формате *PowerPoint* (вы можете настроить сохранение в формате **.ppt** по умолчанию в разделе Параметры, выбрав пункт Загрузка/Сохранение), просто из-за того, что время от времени требуется показывать презентации на компьютере, где поддерживается формат PPT, и гораздо проще работать с тем, что есть.

Существует много абсолютно переносимых типов файлов. Например, вашу коллекцию фотографий, состоящую из большого количества **.jpg**-файлов, может хранить, изменять, редактировать и печатать любое фотоприложение Linux.

Аналогично, музыкальные файлы обычно хранятся в формате MP3, и большинство дистрибутивов проигрывают их при помощи медиаплейеров, например, *VLC*; некоторые дистрибутивы (например, Fedora) потребуют установки дополнительных приложений вроде *LAME*, для преобразования вашей CD-коллекции в MP3. Возможны проблемы, если вы покупаете песни в музыкальном магазине *iTunes* или в центрах распространения музыки для Windows, так как они, вероятно, стеснены некоторыми формами DRM (технические средства защиты авторских прав на цифровые произведения). Самый простой способ преодолеть зыби DRM – либо покупать CD и самим конвертировать »

Скорая помощь

В Великобритании и некоторых других странах загадочные законы приводят к причудам. Фактически копирование музыки с CD в MP3-плейеры стало противозаконным; также во многих странах просмотр содержимого DVD на вашем компьютере относится к области действия серого права. Для проигрывания музыкальных файлов, имеющих кодеки проприетарной природы, пользователям некоторых дистрибутивов Linux необходимо установить следующие программы:

› *LAME* – это MP3-кодировщик для преобразования аудиофайлов в формат MP3. **http://lame.sourceforge.net**

› VLC поддерживает огромное количество типов аудио- и видеофайлов и контейнеров. **www.videolan.org/vlc**

» аудиотреки в MP3, либо покупать более дорогие, но более качественные и свободные от DRM песни в *iTunes* (вы можете обновить любой из ранее купленных треков за небольшую доплату), или же записывать купленные треки на CD, а затем конвертировать их в вашем любимом музыкальном плейере в Linux. Это несколько туманно с точки зрения легальности, но куда важнее то, что последний способ значительно ухудшает качество треков.

Шрифт не фонтан

Еще одна область, чреватая проблемами – это шрифты. Большинство дистрибутивов Linux поставляются с обширным набором шрифтов, использующихся в *OOo* и других приложениях, но среди них может не оказаться тех, которые вы использовали в документах Windows.

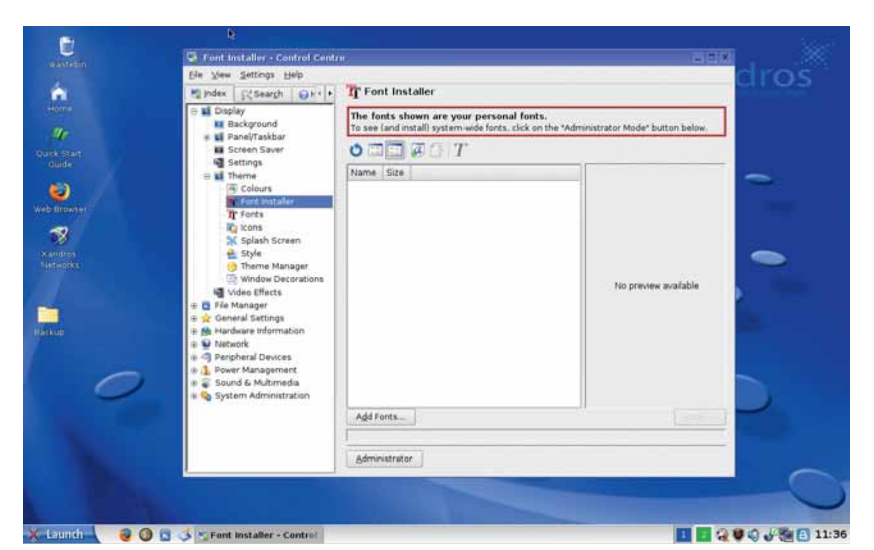
Попробуем перенести шрифты из Windows в Linux. По умолчанию шрифты в Windows хранятся в папке **\windows\fonts**, и здесь вы можете увидеть как шрифты Type 1, так и TrueType, хотя вероятно, что последние более употребимы. Однако знайте, что Windows не утруждает себя хранением всех шрифтов в одном месте. Если вы обнаружите, что какой-то шрифт не содержится в данной папке, перейдите в раздел Шрифты в Панели Управления, выберите в списке нужный вам шрифт, щелкните по его имени правой кнопкой мыши, выберите пункт Свойства и посмотрите полный путь к файлу шрифта. Для этого, естественно, нужно загрузить Windows. Найдя все шрифты, которые вы хотите перенести, скопируйте каждый из них в отдельную папку, находящуюся в папке **Документы**, на разделе Windows. Теперь снова загружайте Linux, с помощью браузеров *Konqueror* или *Nautilus* переходите на раздел Windows, найдите папку со шрифтами и скопируйте ее на ваш рабочий стол в Linux [легальные перспективы этой операции также туманны, – *прим. ред.*].

Скорая помощь

Последняя версия *Opera* под Linux имеет систему *Browser Sync* для перекрестной синхронизации рабочих столов.

Процедуры добавления шрифтов в систему различны для KDE и Gnome; мы начнем с KDE. Щелкните по главной панели и перейдите в Центр Управления KDE. Выберите раздел «Установка шрифтов» (но не «Шрифты»: там настраиваются шрифты для рабочего стола или для меню).

Установщик шрифтов в KDE имеет два режима. Во-первых, можно установить шрифты только для одного пользователя – это иногда полезно, но чаще означает, что будет установлено несколько версий одного и того же шрифта, если другим пользователям он тоже понадобится. Более разумно установить шрифты для всей системы, чтобы все пользователи имели доступ к одному и тому же хранилищу шрифтов: это сэкономит память. Нажмите Режим Администратора, введите пароль root и нажмите на кнопку Добавить шрифты. Теперь найдите шрифты, которые хотите установить, и нажмите Открыть. К счастью, браузер файлов работает так же, как *Windows Explorer*, и если вы хотите установить сразу несколько шрифтов, выберите первый из них в списке и, удерживая клавишу Shift, укажите последний, чтобы выделить и все файлы между ними. Если вы хотите установить несколько отдельных шрифтов из списка, то выделяйте каждый из них, удерживая нажатой клавишу Ctrl. Для добавления шрифтов в систему снова нажмите Открыть.



› В KDE встроено приличное приложение для управления шрифтами. Им можно пользоваться и при работе в Gnome.

Время добавления зависит от скорости вашей системы и количества устанавливаемых одновременно шрифтов, но, как только процесс завершится, шрифты будут доступны для всех ваших приложений.

В Gnome существует два метода установки шрифтов. Первый – использовать приложение *kfontview*. Вам может потребоваться установить его – запустите терминал и введите команду

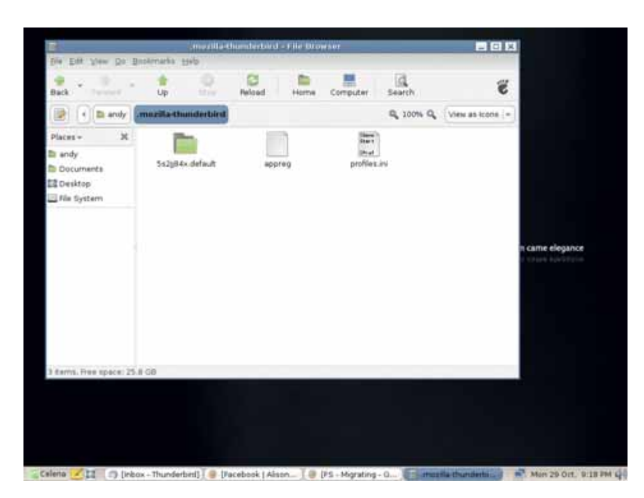
```
sudo apt-get install kontrol
```

Когда необходимые пакеты будут скачаны, а приложение установлено, введите в терминале команду kfontview для запуска приложения. Остальная последовательность действий такая же, как описано выше, поскольку, как вы могли догадаться, этот метод просто устанавливает менеджер шрифтов KDE. Правда, тут создастся новая проблема: заодно установится много другой начинки KDE, и если вы сторонник чистого Gnome или у вас мало свободного места, то уж лучше добавить шрифты вручную Для этого откройте *Nautilus* и пройдите в каталог **/home/имя_пользователя**. Выберите пункт меню Вид -> Показывать скрытые файлы; вы узнаете их, потому что все они начинаются с точки. Шрифты должны храниться в папке **.fonts**, так что откройте ее или, если ее пока нет, создайте новую папку и назовите ее **.fonts**. Другой способ найти эту папку – набрать в адресной строке *Nautilus* **fonts:///** и нажать кнопку Перейти.

Теперь для установки шрифтов просто перетащите файлы в эту папку. Однако, имейте в виду, что *Nautilus* может не показать файлы сразу после того, как вы их перетащили: шрифтам, чтобы стать доступными, нужно сначала активироваться Завершите сеанс, затем войдите снова, тогда шрифты будут доступны для установленных приложений.

Электронная почта

Хотя службы мгновенных сообщений и Facebook, кажется, отнимают первенство у электронной почты, последняя все еще занимает львиную долю трафика в сети. Для многих пользователей это и средство общения, и блокнот, и способ создания резервных копий документов. А значит, важно удостовериться, что эта часть перехода совершена правильно. В Windows много почтовых программ, но мы остановимся на двух наиболее популярных: *Outlook Express* от Microsoft и *Thunderbird* от *Firefox*. Перенос данных из *Thunderbird* под Windows в *Thunderbird* под Linux тривиален, поэтому займемся переносом данных из почтового клиента MS. Начинается эта процедура с... установки *Thunderbird* под Windows. Скачайте последнюю версию *Thunderbird* для Windows с сайта проекта Mozilla и установите ее. При запуске приложения вы сможете импортировать все учетные записи из почтового клиента, используемого по умолчанию (в нашем случае это *Outlook Express*). Сделав это, закройте Windows и загрузитесь под Linux. Если в разделе Linux установлен *Thunderbird*, необходимо определить, где он хранит почту. Скорее всего это каталог **.Thunderbird** или **.mozilla_Thunderbird** в вашем домашнем разделе (помните: точка перед именем означает, что каталог скрытый, для его просмотра перейдите в меню Вид > Показывать скрытые файлы). Теперь, используя или *Konqueror*, или *Nautilus*, перейдите в раздел Windows – его точное расположение зависит от дистрибутива Linux, который вы используете, но поищите в папках с названием **/disks**, **/mnt** или **/media** – и найдите папку *Thunderbird*. Она должна быть в **C:\Documents and Settings\username\Application Data\Thunderbird**, то есть нужно искать, например, здесь: **/disks/Windows/Documents and Settings/username/Application Data/Thunderbird**, поскольку в Windows для обозначения директорий используется обратный слэш, а в Linux – прямой. Затем просто перенесите содержимое папки из Windows в папку **.thunderbird** раздела Linux. Теперь закройте браузер и запустите *Thunderbird*, чтобы убедиться, что вся ваша почта и настройки стали такими же, как в Windows. Обратитесь к руководству на стр. 70 для более глубокого изучения переноса всех настроек *Outlook* в ваш дистрибутив Linux.



› Настройки у *Thunderbird* кросс-платформенные, и перенос электронной почты с Windows на Linux – ерундовое дело.

› Синхронизация браузеров от Google полезна не только при переходе на другую систему: она пригодится тем, кто работает на нескольких компьютерах.

Последний вариант – войти на IMAP-совместимый почтовый web-сервис. Он позволит перенести всю вашу почту в Web, после чего вы можете установить любой почтовый клиент для Linux, поставить сервис IMAP, и все ваши данные будут отображены снова в локальный почтовый клиент. Это может быть долгий и витиеватый процесс (у нас нет места для его описания), особенно если у вас тысячи сообщений, но преимущество в том, что вы создаете резервную копию всей истории сообщений, доступ к которой возможен с любой машины, и при этом продолжаете использовать ваш любимый почтовый клиент. Google, очевидно, добавил поддержку IMAP для Gmail, но есть и другие сервисы, поддерживающие эту функцию.

Настройки браузера

Можно использовать тот же трюк для переноса настроек браузера из Windows-версии *Firefox*'а в Linux версию. Однако разработчики Google сделали небольшое расширение, которое можно использовать для синхронизации между компьютерами содержимого и настроек *Firefox*'а – включая cookies, сохраненные пароли, закладки и историю посещений. Это удобно, если у вас несколько компьютеров (доступно все, независимо от компьютера, за которым вы сидите), а еще удобнее при переходе с одной ОС на другую.

Аналогично всему вышеупомянутому, переход состоит из двух стадий. Но в этом случае обе они одинаковые.

Зайдите из Windows на **www.google.com/tools/Firefox/browsersync/** и щелкните по ссылке Install Now. Когда вы согласитесь с условиями, будет установлено расширение для вашего браузера. Далее для запуска системы вам потребуется перезапустить *Firefox*. В последующих диалоговых окнах вам предложат завести учетную запись на Google, или ввести ваши данные, если она у вас уже есть, и выбрать PIN из четырех цифр – он будет использоваться для шифрования и защиты всех персональных данных, выгруженных на сервера Google. После того, как это будет сделано, приложение определит ваши настройки *Firefox* и отошлет их на сервер. После завершения можете перейти в Linux.

Здесь процесс такой же, но мы считаем за благо перед запуском XPI очистить закладки новой системы Linux, поскольку они полны всякой ерунды, зависящей от дистрибутива, которая вряд ли вам понадобится (вы, конечно, можете оставить то, что вам понравилось), а все это попадет в ваш профиль Browser Sync. Когда вы запустите расширение, вас снова спросят имя пользователя на Google, пароль и ваш PIN. Приложение начнет скачивать параметры с web и применять их к той версии *Firefox*, что запущена у вас. В зависимости от объема истории, процесс может занимать до 20 минут, но это действие однократное. По завершении, Linux-версия *Firefox* станет клоном вашей Windows-версии; просто начните вводить адрес, и в адресной строке вы увидите предыдущие записи, как если бы они ранее вводились из-под Linux. Ваши закладки также станут доступными в своих исходных папках, все сохраненные имена пользователей и пароли будут вводиться автоматически.

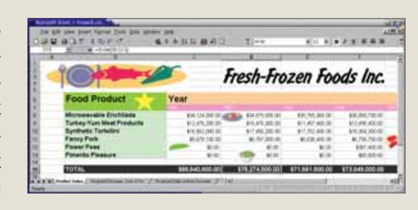
Еще круче то, что после закрытия *Firefox* изменения отправляются в ваш онлайн-профиль и хранятся там, чтобы при следующей загрузке *Firefox* – под Windows, Linux или OS X, это был бы тот же самый браузер, и ничего не было бы потеряно.

На нашем уроке мы рассмотрели несколько способов облегчить переход с Windows на Linux. Сейчас, когда Vista вызывает столько проблем для разработчиков драйверов, и все приближающимся концом Windows XP, самое время исследовать альтернативные возможности. LXF

Не переходи — виртуализируйся

Если вы – принципиальный пользователь *Word/Excel*, то *OOo*, наверное, смутит вас, когда вы попытаетесь перенести ваши волнующие произведения в новые приложения. Ну так используйте либо виртуализацию, либо эмуляцию! Вы получите все преимущества Linux как ОС, сохраняя доступ к вашим любимым Windows-приложениям. В первом случае создайте в вашей Linux-системе виртуальную машину, используя либо проприетарные приложения типа *VMware* или *Parallels*, либо их открытые аналоги от *Xen*, *KVM* или других, затем установите на машину полную версию Linux. Работайте себе под Linux, а когда понадобится запустить *Word*, *Excel* или что-нибудь еще, запустите виртуальную машину с Windows и получите доступ к тем приложениям, как будто бы вы работаете на обычном ПК с Windows.

Если это кажется вам стрельбой из пушки по воробьям, то есть возможность запускать открытую реализацию программных интерфейсов приложений (API) Windows, которая называется *Wine*, или ее коммерческую ветвь *Crossover Pro*. Производительность этого метода обычно выше, чем при виртуализации, поскольку нет накладных расходов на ОС, но он умеет работать не со всеми приложениями. Последняя версия *Crossover Pro* поддерживает версии *Microsoft Office* вплоть до 2003, но надежной поддержки *Office 2007* в течение нескольких месяцев (или даже лет) не предвидится, так что потерпите...

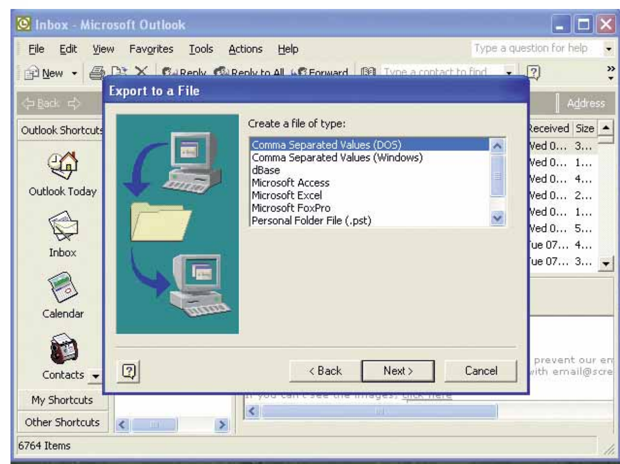


› Really Simple Syndication расскажет, что творится в мире.

Экспорт: календарь,

ЧАСТЬ 1: Экспорт ваших контактов из *Outlook* или *Outlook Express* может стать настоящей головной болью. **Джо Касселс** объяснит, как ее избежать.





› Экспорт списка контактов из *Outlook* в формате .csv (значения, разделенные запятыми) намного упростит жизнь.



Наш эксперт

Джо Касселс потратил массу времени, экспериментируя с различными способами помочь людям выбраться из темноты.

Сейчас на свете полно простых в использовании дистрибутивов Linux, и мигрировать из Windows не так уж и сложно. Перенести данные также легко. Если у вас конфигурация с двойной загрузкой, создайте общий раздел для данных; в противном случае проще всего скопировать данные из одной системы в другую с помощью USB-брелка, так как не придется иметь дело с сетевыми правами доступа.

Если вы какое-то время пользовались Windows-версиями популярных приложений с открытым исходным кодом, то, начав работать в KDE или Gnome, культурного шока не испытаете. Обе среды ближе к Windows, чем Mac OS. Зато пользователей *Outlook* ожидает реальная головная боль. Перенос персональных данных из файлов закрытого Microsoft формата PST – дело, мягко говоря, непростое. Нельзя просто скопировать PST-файл в Linux и импортировать его в *Kontact*, *Evolution* или *Thunderbird*: он не будет работать. Этот файл запутан покруче иного морского узла – классический пример «иглы одного поставщика».

Если этот файл нельзя открыть нигде, кроме *Outlook*, а *Outlook* нельзя запустить нигде, кроме Windows – считай, при переходе мы хорошо завязли. Предстоит сделать выбор: похоронить свои данные, убедиться, что *Outlook* запускается на виртуальной машине Windows, или продолжать использовать раздел Windows. Любому, кто собрался послать Windows прощальный поцелуй, придется крепко подумать.

Экспорт из Windows

Осознаем ключевой момент: переезд требует планирования. Пока вы в Windows, нужно предпринять шаги, обеспечивающие экспорт данных в формат, с которым сможет работать новая программа.

Решить эту проблему можно несколькими способами. *Outlook* может экспортировать данные в файлы разных форматов, но для нас действительно полезен только один из них – CSV ('comma-separated values' – «значения, разделенные запятыми»). Внутренность папки преобразуется в простой текст, содержащий поля, разделенные запятыми. Это хороший вариант для списка контактов и других небольших баз данных, но он провалится, если попытаться экспортировать архив электронной почты. Если вы пользуетесь почтовым сервером IMAP вместо POP, то сможете считать оттуда все сообщения, так как не храните их локально, но проблема с сообщениями в архиве останется.

Календарь

События календаря можно экспортировать из *Outlook* в файл формата CSV, но в данном случае это не лучший выбор. Гораздо практичнее конвертировать календарь в стандартный формат ICS, используемый web-календарями.

В *Outlook 2007* есть возможность выслать календарь по электронной почте в файле формата ICS. ICS-файл легко импортируется в *Evolution* или любую другую программу-календарь. Чтобы сделать это, откройте *Outlook* и переключитесь на календарь. На панели навигации выберите Отправить Календарь По Почте [Send Calendar Via Email]. Задайте диапазон дат, который нужно экспортировать, и другие необходимые настройки. После завершения работы мастера будет создано электронное сообщение с прикрепленным к нему файлом ICS. Теперь сохраните вложение на жесткий диск или другой носитель, с помощью которого вы будете передавать данные в Linux.

В более ранних версиях *Outlook* экспортировать данные непосредственно в ICS нельзя, поэтому нам понадобится макрос, который сделает эту работу. Загрузить макрос *Outlook2iCal* можно с сайта **http://outlook2ical.sourceforge.net**, причем на ваш ПК с Windows, поскольку файл надо будет распаковать и установить в *Outlook*. Но сперва позаботьтесь, чтобы *Outlook* был не против принять макросы. Для этого откройте Сервис > Макросы > Безопасность [Tools > Macro > Security] и измените уровень безопасности на Средний [Medium] или более низкий.

Для установки макроса откройте редактор Visual Basic. Выберите Сервис > Макросы > Редактор Visual Basic [Tools > Macro > Visual Basic Editor]. В окне проекта слева кликните правой кнопкой мыши на икон-

Почта и контакты

ке проекта в верхней части дерева и выберите «Импорт файла» (Import File). Найдите каталог, куда вы распаковали *Outlook2iCal*, и выберите файл **frmSaveiCalStep1.frm**. Сделайте то же самое с другими файлами в этом же каталоге: **frmSaveiCalStep2.frm**, **frmSaveiCalStep3.frm**, **clsFTP.cls** и **icalendar.bas**. Закройте редактор Visual Basic.

Теперь можете запустить макрос, выбрав Tools > Macro > Macros > export_icalendar > Run. Макрос запустит несложного Мастера. Выберите календарь, который хотите экспортировать, и нажмите Далее [Next]. Если календарь у вас большой, придется немного подождать. При наличии в календаре категорий, на этой стадии вы сможете отобрать из них только нужные, либо выбрать Export All Appointments In Calendar [Выбрать все]. Сохраните календарь на диске, задав ему имя и местоположение. Нажмите Завершить [Finish]. Теперь календарь можно скопировать на носитель, предназначенный для переноса данных.

Почта

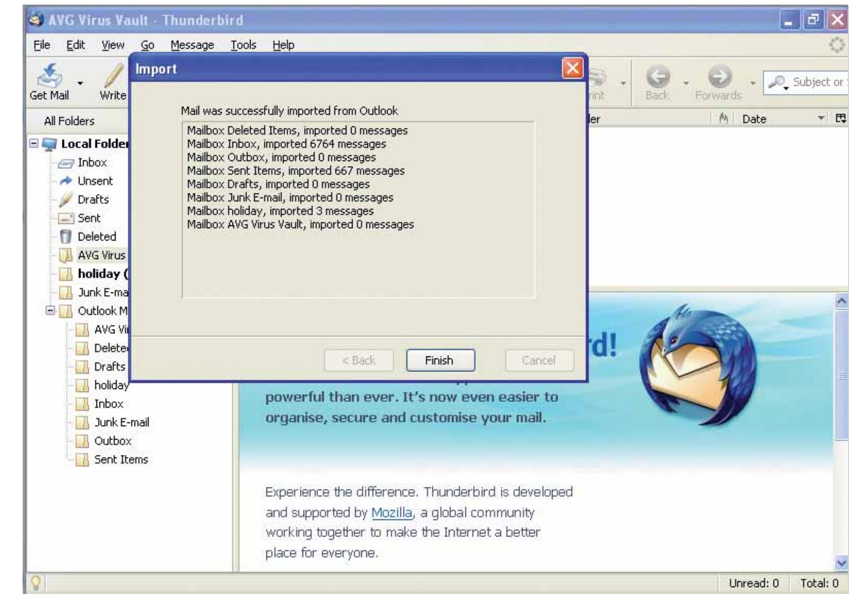
Сложнее всего экспортировать почтовый архив, особенно если он рассортирован по папкам, а некоторых сообщений нет ни в одной папке *Outlook*. Прежде чем начать, создайте каталог для таких сообщений и перенесите их туда. Вы также обнаружите, что чем больше файл PST, тем выше вероятность возникновения проблем с его экспортом, и лучше уменьшить количество сообщений до начала процесса. Сейчас самое время найти сообщения, которые вряд ли понадобятся снова и просто занимают место, и удалить их. Небольшое «обезжиривание» повысит шансы на успех.

Outlook не экспортирует данные в какой-либо из открытых почтовых форматов, и нам потребуется программа-посредник. Другие программы, запущенные в той же системе, что и *Outlook*, могут импортировать почту непосредственно из *Outlook*, так как у них есть доступ к библиотекам для декодирования PST-файлов. Это означает, что данные из Outlook можно перенести в Windows-версию *Thunderbird*, конвертируя их в открытый формат.

Загрузить *Thunderbird* для Windows можно с сайта http://www.mozilla-russia.org/products/thunderbird/. Установите ее в Windows обычным образом. При первом запуске *Thunderbird* будет предложено импортировать данные и настройки. Выберите «Не импортировать ничего на этом этапе [Don't Import Anything At This Stage]. Также не



› Импортируйте почтовые сообщения с помощью *Mozilla Thunderbird* для Windows.



› Если *Thunderbird* импортировал почтовые папки, значит, они пригодны для переноса в Linux.

нужно делать *Thunderbird* почтовой программой по умолчанию. Потом можно будет импортировать сообщения поэтапно, если потребуется.

В *Thunderbird* выберите пункт меню Сервис > Импорт [Tools > Import] и затем Почта [Mail]. Нажмите Далее [Next] и выберите импорт данных из *Outlook*. Снова нажмите Далее», чтобы начать импорт. *Thunderbird* сконвертирует почтовые папки *Outlook* и после завершения импорта покажет результат. Полюбопытствуйте в какой-нибудь из папок, правильно ли были импортированы сообщения или нет. Если все хорошо, закройте *Thunderbird*, так как теперь можно просто скопировать почтовые папки, которые он использует. Обычно они находятся в каталоге **C:\Documents and Settings\имя_пользователя\Application Data\Thunderbird\Profiles\идентификатор_профиля\Mail\Local Folders** в Windows XP и в каталоге **C:\Users\имя_пользователя\AppData\Local\Thunderbird\Profiles\идентификатор_профиля\Mail\Local Folders** в Vista. Чтобы увидеть каталог **AppData**, нужно включить отображение скрытых файлов и каталогов. Для этого в XP откройте папку и выберите Сервис > Свойства папки > Вид > Показывать скрытые файлы и папки [Tools > Folder Options > View > Show hidden files and folders]. В Vista быстрее всего это можно сделать, напечатав Folder Options в строке поиска меню «Пуск». Скопируйте содержимое папки **Mail** целиком на диск для переноса данных в Linux.

При импорте большого объема писем в *Thunderbird* могут возникнуть проблемы. В этом случае полезно попробовать экспортировать данные порциями, каждый раз сохраняя копию файла PST и удаляя все сообщения, которые не входят в заданный диапазон дат. Оставшиеся сообщения затем можно импортировать в *Thunderbird* и скопировать файлы таким же образом, как и раньше. После этого нужно вернуться в *Outlook*, импортировать PST-файл из резервной копии и повторить эти операции с другим диапазоном дат. Воспользуйтесь этим подходом, если импортировать архив целиком не удается.

Контакты

Самый простой способ перенести контакты – экспортировать их в CSV-файл из *Outlook*. Выберите пункт меню Файл > Импорт и Экспорт > »

Ресурсы

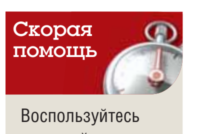
› *Outlook2iCal*
Утилита экспорта для ранних версий *Outlook*, не поддерживающих экспорт в формат iCal.
http://outlook2ical.sourceforge.net

› *Mozilla Thunderbird*
Версия для Windows. Нужна для экспорта почтовых архивов *Outlook* в формат, который могут обрабатывать программы для электронной почты в Linux. http://www.mozilla-russia.org/products/thunderbird/

› *Mozilla Sunbird*
Версия для Linux
http://www.mozilla-russia.org/projects/calendar/sunbird/

Скорая помощь

При экспорте данных из файла ICS диапазон дат ограничен десятью годами. Для повторяющихся событий, вроде дней рождения, которые выпадают из этого диапазона, может потребоваться изменение их повторяемости при импорте в *Evolution*.



Скорая помощь

Воспользуйтесь утилитой экспорта *Outlook* для сохранения резервной копии PST-файла. Это поможет восстановить данные, если вы случайно удалите не те сообщения или угробите данные в процессе переноса. Записав эту копию на DVD, вы будете уверены, что данные в сохранности, пусть и в закрытом формате. В чрезвычайной ситуации вы можете воспользоваться компьютером друга с ОС Windows.

» Экспорт в файл > Разделенные запятыми значения (DOS) > Контакты [File > Import and Export > Export to a file > Comma Separated Values (DOS) > Contacts] и укажите имя файла и каталог, в который он будет сохранен. После создания файла скопируйте его на диск для переноса данных.

Альтернатива – импортировать контакты *Outlook* в Windows-версию *Thunderbird* и затем экспортировать их оттуда. Этот подход удобен, когда файл CSV не импортируется должным образом в новую почтовую программу. В *Thunderbird* выберите Сервис > Импорт > Адресные книги > Outlook [Tools > Import > Address Books > Outlook] и подождите, пока импортируются контакты. Затем нажмите кнопку Адресная книга [Address Book] на панели инструментов и выберите адресную книгу, предназначенную для экспорта. Выберите пункт меню Сервис > Экспорт [Tools > Export] и укажите имя файла. Оставьте тип адресной книги по умолчанию (LDIF) и нажмите кнопку Сохранить [Save]. Теперь можно скопировать этот файл на диск с остальными файлами.

Задачи

С задачами будет посложнее, потому что, в отличие от календарей, единого формата файлов для них нет. Как и список контактов, задачи можно экспортировать в файл формата CSV. К несчастью, этот файл не получится импортировать ни в *Evolution*, ни в *Kontact*, что немного раздражает. Впрочем, можно распечатать список контактов из *Outlook*, чтобы потом было проще снова вводить их вручную.

Импорт в Linux

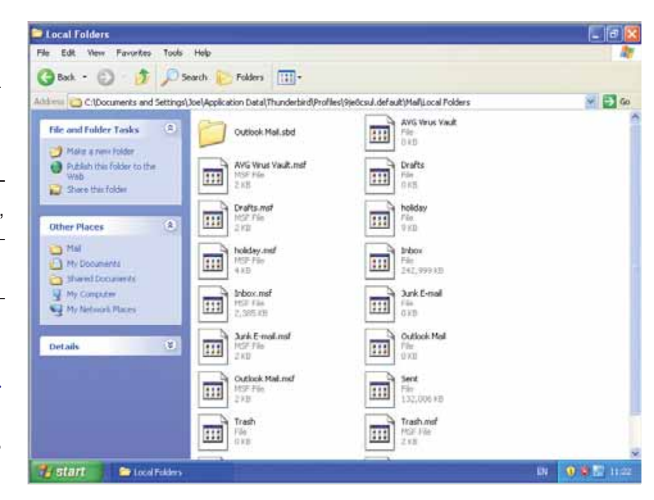
Экспортировав календарь, сообщения электронной почты, контакты и задачи в лучших возможных для *Outlook* форматах, можете переносить их в Linux и импортировать в подходящую программу. Подключите диск для переноса данных или смонтируйте CD или DVD с ними и скопируйте данные в ваш домашний каталог. Можно этого и не делать, но гарантия того, что данные сохранены локально, поможет существенно упростить все, что мы будем делать дальше.

Контакты

Начнем с самого простого. Откройте *Evolution* и нажмите на кнопку Контакты [Contacts]. Выберите Файл > Импорт [File > Import]; запустится Мастер Импорта Evolution [Evolution Import Assistant]. Нажмите Вперед [Forward] и выберите Импорт из файла [Import a single file]. Нажмите Вперед [Forward], затем кнопку Имя файла [Filename]. Выберите CSV-файл, содержащий данные о контактах. Нажмите Вперед [Forward] и укажите каталог, куда будут импортированы данные (скорее всего, персональные папки). Нажмите Вперед [Forward], затем Импорт – и дело сделано. Просмотрите некоторые контакты, чтобы убедиться в правильности импорта. Если в CSV-файле есть ошибки сдвига строки или поля, не распознаваемые *Evolution*, могут возникнуть проблемы.

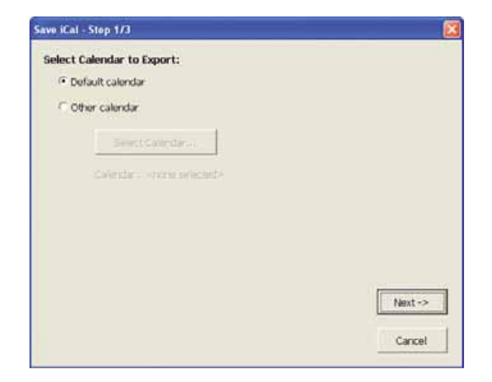
Если импорт не удался, откройте CSV-файл с контактами в редакторе электронных таблиц и сравните содержимое полей с контактами, чтобы понять причину ошибки. Попробуйте отредактировать файл таким образом, чтобы поля в нем появлялись в том же порядке, в котором они импортируются *Evolution*. Сохраните копию файла под другим именем и удалите все контакты в *Evolution*. Импортируйте измененную версию CSV-файла. Если счастья все нет, попробуйте сгенерировать адресную книгу в формате LDIF с помощью Windows-версии *Thunderbird* и импортируйте в *Evolution* ее.

Если вы используете *Kontact*, запустите программу и нажмите на кнопку Контакты [Contacts]. Выберите Файл > Импорт > Импорт файла CSV [File > Import > Import CSV list] и укажите файл с контактами. Здесь можно столкнуться с теми же проблемами, что и в случае с *Evolution*, и если импорт данных из CSV не получается, решением опять будет импорт из LDIF-файла. Если вы предпочитаете *Thunderbird*, то там импортировать контакты очень просто. Запустите программу и выберите Сервис > Импорт > Адресные книги [Tools > Import > Address Books] и укажите файл, который хотите импортировать. Для лучшей совместимости можно пропустить контакты *Outlook* через *Thunderbird* для Windows, а уж потом переносить их в Linux.



› **Скопируйте файлы, содержащие почтовые папки, на носитель.**

› Лучше всего конвертировать события календаря в файл формата ICS; тут поможет *Outlook2iCal*.



Календарь

Теперь импортируем ваши встречи. В *Evolution* переключитесь на календарь и выберите Файл > Импорт [File > Import]. Запустится мастер импорта. Нажмите Вперед [Forward] и выберите Импорт из файла [Import a single file]. Нажмите кнопку рядом с именем файла и выберите свой файл формата ICS. Нажмите Открыть [Open], затем Вперед [Forward] и выберите календарь, который хотите импортировать. Нажмите Импортировать [Import] для выполнения импорта.

В *Kontact* переключитесь на календарь и выберите пункт меню Файл > Импорт > Импортировать календарь [File > Import > Import Calendar].

> «OUTLOOK НЕ ЭКСПОРТИРУЕТ ПОЧТОВЫЕ ПАПКИ В ОТКРЫТЫЙ ФОРМАТ: ЭТО СДЕЛАЕТЕ ВЫ.»

Найдите ICS-файл и нажмите Открыть [Open]. Выберите Объединить с текущим календарем [Merge Into Existing Calendar] и нажмите OK.

Если вы хотите работать с почтой и контактами в *Thunderbird*, то знайте, что календаря в нем нет. Поэтому вам может пригодиться программа Mozilla *Sunbird*, найти которую можно на странице **http://www.mozilla-russia.org/projects/calendar/sunbird/**. Загрузите версию под Linux и распакуйте содержимое архива в свой домашний каталог. В окне терминала перейдите в каталог *Sunbird* и введите команду *sh sunbird* для запуска программы. Выберите Файл > Импорт [File > Import], найдите ICS-файл, содержащий календарь, и выберите Открыть [Open], чтобы импортировать его.

Почта

Экспортировав почтовые папки с помощью *Thunderbird* для Windows, импортировать их в Linux на удивление просто: нужно всего лишь скопировать файлы в правильное место. Откройте свой домашний каталог и выберите Вид > Отображать скрытые файлы [View > Show hidden files]. Откройте каталог **.evolution**, а затем – почтовый каталог и локальный каталог. Туда-то и нужно скопировать файлы почтовых папок. Откройте новое окно для просмотра экспортированных файлов. Скорее всего, вы найдете их в папке **Local Folders**, куда бы вы их ни копировали. С каждой почтовой папкой связано два файла; оба они называются так же, как и папка, но у одного есть расширение **MSF**.



› Импортируйте свой список контактов в формате CSV в *Evolution*, чтобы создать адресную книгу под Linux.

Например, для папки Входящие [Inbox] есть файлы **inbox** и **inbox.msf**. Скопируйте эти файлы в локальный каталог в **.evolution/mail**. Сделайте то же самое для остальных каталогов, которые нужно импортировать. Откройте *Evolution* и нажмите на кнопку Почта [Mail], и вы увидите импортированные папки и электронные сообщения.

В *Kontact* все примерно так же. Нужно найти каталог **~/.kde/share/apps/kamil/mail**, где **~/i** – ваш домашний каталог. Как и для *Evolution*, скопируйте туда почтовые папки. Запустите *Kontact* и переключитесь на вкладку Почта [Mail]. Там вы увидите свои почтовые папки и сообщения. В *Thunderbird* процедура практически аналогична. Откройте каталог **~/.mozilla-Thunderbird/идентификатор_профиля/Mail/Local Folders**, скопируйте туда почтовые папки и запустите *Thunderbird*, чтобы проверить, что все в порядке. Помните, что папки, имена которых начинаются с точки, вы увидите, только включив показ скрытых файлов. Если все подготовительные действия в Windows были правильными, то импорт почтовых папок в Linux – это всего лишь копирование файлов в правильное место.

Задачи

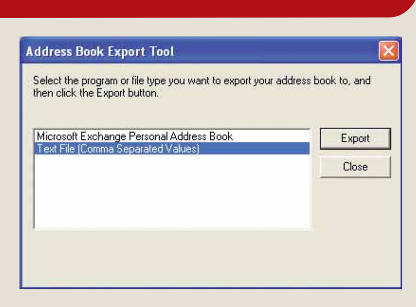
К сожалению, список задач, он же To Do list, похоже, придется создавать заново. И это очень обидно, потому что теории он должен быть небольшим, но я еще не придумал, как надежно перенести данные из *Outlook* в другую программу. Экспортировав список задач в CSV-файл, можно открыть его в редакторе электронных таблиц и скопировать и вставить содержимое каждой записи, но это не очень удобно, так как каждое поле нужно копировать и вставлять отдельно. Может, проще просто распечатать этот список и потратить некоторое время на его ручной ввод.



› Для импорта почтовых папок в *Evolution* скопируйте почтовые файлы Thunderbird в каталог ~/.evolution/mail/local.

Outlook Express

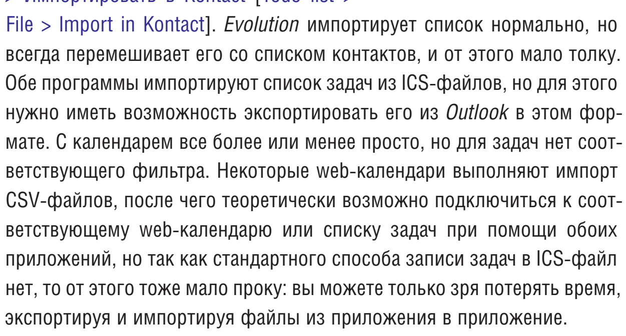
Хотя *Outlook Express* и не использует файлы формата PST, проблем с ним не меньше. Выбрав Файл > Экспорт > Сообщения [File > Export > Messages], можно экспортировать сообщения электронной почты, но только в *Outlook* или *Microsoft Exchange* – ни то, ни другое вас не спасет. Но можно воспользоваться тем же приемом, что и с *Outlook*. В *Thunderbird* выберите Сервис > Импорт [Tools > Import] и затем Почта [Mail]. Затем кликните Дальше [Next] и выберите Импорт данных из Outlook Express [Import from Outlook Express]. Почтовые папки будут преобразованы в формат *Thunderbird*: скопируйте их на диск для переноса данных. Адресную книгу *Outlook Express* можно экспортировать, выбрав пункт меню Файл > Экспорт > Адресная книга > Текстовый файл (Разделенные запятыми значения) > Экспорт [File > Export > Address Book > Text File (Comma Separated Values) > Export]. Укажите имя файла и каталог и завершите работу мастера.



› В *Outlook Express* можно экспортировать адресную книгу в файл формата CSV.

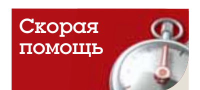


И в *Evolution*, и в *Kontact* есть возможность импортировать список задач. В *Evolution* переключитесь на список задач и выберите Файл > Импорт [File > Import]. Откроется мастер импорта. В *Kontact* выберите Список задач > Файл > Импортировать в Kontact [Todo list > File > Import in Kontact]. *Evolution* импортирует список нормально, но всегда перемешивает его со списком контактов, и от этого мало толку. Обе программы импортируют список задач из ICS-файлов, но для этого нужно иметь возможность экспортировать его из *Outlook* в этом формате. С календарем все более или менее просто, но для задач нет соответствующего фильтра. Некоторые web-календари выполняют импорт CSV-файлов, после чего теоретически возможно подключиться к соответствующему web-календарю или списку задач при помощи обоих приложений, но так как стандартного способа записи задач в ICS-файл нет, то от этого тоже мало проку: вы можете только зря потерять время, экспортируя и импортируя файлы из приложения в приложение.



Единственный выход в том случае, если вы не хотите вводить данные вручную в *Evolution*, *Kontact* или *Sunbird* – воспользоваться web-ориентированным списком задач. Проект *Remember The Milk* (**www.rememberthemilk.com**) предоставляет список задач, в который можно очень просто импортировать свои задачи. Зайдите на сайт и создайте учетную запись (бесплатно). После заполнения необходимых полей и подтверждения регистрации можно отправить список задач по электронной почте. Зайдите в свой профиль и кликните на ссылке Settings в правом верхнем углу. Затем выберите вкладку Info. На ней вы найдете электронный адрес для импорта сообщений (Import Email Address). Скопируйте его и создайте новое письмо на этот адрес. В теме письма введите имя файла списка, в котором будет отправлен список задач. Откройте свой CSV-файл в редакторе электронных таблиц и скопируйте столбец, содержащий названия задач. Вставьте этот столбец в текст письма и отправьте его. Список задач будет импортирован на сайт, но даты и времена все-таки будут отсутствовать. LXF

› Можно импортировать календарь в формате ICS в Evolution или связать его с web-календарем.



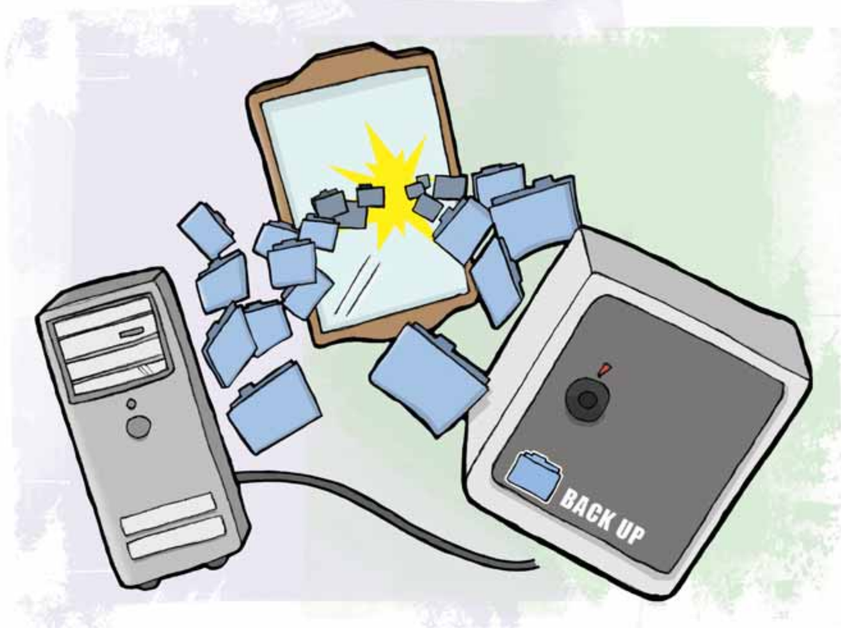
Скорая помощь

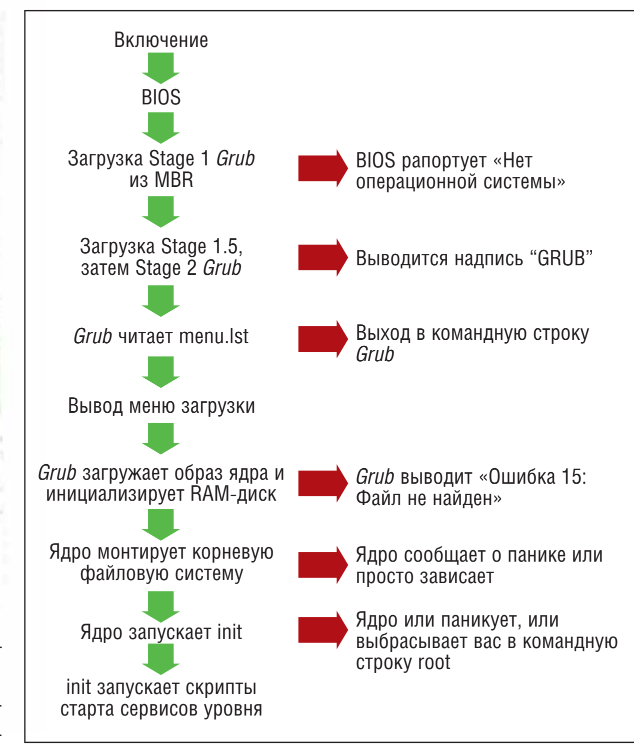
Проверить содержимое CSV-файла можно в редакторе электронных таблиц *OpenOffice.org*. Это поможет вам определить, не сбились ли в файле границы полей, из-за чего данные могут распознаваться неправильно. Если вы решили произвести изменения в файле, сохраните его под другим именем и убедитесь, что также сохранили его в CSV.

» **Через месяц** Синхронизируйте данные между Windows и Linux по сети.

Когда Linux не

Мысль об этом вгоняет в дрожь пользователей всех ОС - не дай Бог, компьютер откажется загружаться. **Д-р Крис Браун** прольет бальзам на ваши раны.





› Рис. 1: Нормальный порядок событий загрузки Linux.



Наш эксперт

Д-р Крис Браун внештатный консультант по Linux со степенью доктора наук по физике элементарных частиц и обладатель сертификатов Novell CLP и Red Hat RHCE. Он только что написал книгу по SUSE Linux для O'Reilly.

Для нас, старых волков, загрузка, или «начальная загрузка» – это та глубоко таинственная последовательность операций, выполняемых компьютером между его включением и выводом приглашения войти в систему. За это время куча непонятных сообщений проплывает по экрану, но обычно пользователи их игнорируют, и большинство дистрибутивов Linux скрывает их за красивым экраном с обнадеживающим прогресс-индикатором. И это замечательно – пока работает. На данном уроке мы рассмотрим процесс загрузки более подробно: в частности, выясним, что может пойти неправильно и как диагностировать и устранить проблему.

Проникаемся проблемой

Когда я преподавал Linux на одном из моих курсов, многие слушатели говорили мне, что интересуются устранением неисправностей в той или иной форме. Некоторые ищут рецептов типа кулинарных – «Увидев сообщение об ошибке X, запустите команду Y», но таким способом устранить неисправность удается редко. Мой первый совет всем тем, кто столкнулся с проблемой, всегда один и тот же: «При устранении неисправности первым делом надо понять, как система должна работать. Вторая важная вещь – точно представить, что система пыталась сделать, когда неисправность возникла».

Помня об этом, взглянем на загрузку Linux. Знание нормальной последовательности событий и установление, как далеко она зашла перед возникновением проблемы, являются ключевыми для диагностики и исправления проблем загрузки. Рисунок 1 справа вверху показывает нормальную последовательность событий (зеленые стрелки) и указывает на некоторые возможные пути неудачи (красные стрелки).

Знакомимся с загрузкой поближе

Загрузка – операция многоступенчатая. При включении компьютера контроль вначале переходит к программе, называемой BIOS, которая хранится в постоянной памяти на материнской плате. BIOS выполняет тестирование оборудования и разведку окрестностей на предмет устройств для загрузки, и выводит экран конфигурации, позволяющий установить порядок поиска загрузочных устройств; в современных BIOS их диапазон весьма широк, в том числе PXE-загрузка с сетевого сервера. На нашем уроке мы рассмотрим только загрузку с жесткого диска.

Затем BIOS считывает главную загрузочную запись (Master Boot Record, MBR) с выбранного устройства и запускает ее. (Не найдя MBR, BIOS выведет сообщение вида «Нет операционной системы» и впадет в полный ступор.) MBR занимает первый сектор диска. Он содержит таблицу разделов (64 байта) и очень короткую (446 байт) программу первичной загрузки, или 'stage 1'. Этот загрузчик всего-навсего выводит на экран слово *Grub* и передает эстафету загрузчику второй стадии, используя «карту блоков», внедренную в MBR (карта содержит номера блоков, где находится загрузчик 'stage 2'). Здесь я подразумеваю, что вы используете загрузчик *Grub*. Существует также более старый загрузчик Lilo, но *Grub* «умнее» и используется в большинстве современных дистрибутивов Linux. Второй этап работы загрузчика *Grub* по

» **Месяц назад** Ваша потеря в Linux - наша находка.

загружается

сути дела «полуторный», и если вы посмотрите в директории **/boot/**
Grub, то увидите там файлы, содержащие различные его варианты, с именами вида **e2fs_stage_1_5** и **reiserfs_stage_1_5**. Каждая из этих программ способна получить доступ к файлам по их имени, используя определенный формат файловой системы. **e2fs_stage_1_5** может читать файловые системы ext2 и ext3, **reiserfs_stage_1_5** – файловую систему reiser, и т.д. *Grub* способен обратиться к файлам по их имени во время начальной загрузки (перед запуском Linux), и этим он отличается от Lilo. Программа «полуторного» этапа загружает *Grub* stage 2, который значительно больше. На этом этапе считывается файл конфигурации *Grub* (обычно **/boot/grub/menu.lst** или **/boot/grub/grub.conf**) и, на основе содержащихся там записей, выводится меню выбора операционных систем для загрузки. Если *Grub* не может найти свой файл конфигурации, он переходит в интерактивную командную строку, позволяющую ввести команды *Grub* вручную. Типичное содержание записи **menu.lst** такое:

```
title openSUSE 10.2
    root (hd0,0)
    kernel /boot/vmlinuz-2.6.18.2-34-default root=/dev/hda1 vga=0x317
showopts
    initrd /boot/initrd-2.6.18.2-34-default
```

Первая строчка просто содержит текст, который появится в меню загрузки. Следующие за ней строки содержат команды, которые *Grub* должен выполнить, если вы выберите этот пункт меню. Строка root подсказывает *Grub*, где искать корневую файловую систему. У *Grub* своя система именования разделов диска, отличная от системы, принятой в Linux, что сбивает с толку. На языке *Grub* hd0 означает первый диск – а на обычном ПК с IDE-дисками это будет ссылка на Linux-устройство **/dev/hda**, или, на более свежих дистрибутивах, **/dev/sda**. На языке *Grub* (hd0,0) ссылается на первый раздел этого диска – в Linux это будет **/dev/hda1** или **/dev/sda1**. Строка kernel определяет файл, который *Grub* должен загрузить как ядро Linux; в конце этой строки вы увидите несколько дополнительных параметров загрузки, передаваемых ядру. Подробнее об этом чуть позже. Строка initrd определяет файл 'initial RAM Disk' – образ файловой системы, которая будет использоваться ядром как загрузочная. *Grub* также отвечает за загрузку его в память. Если *Grub* не может найти ядро или образ ram-диска, он выводит Error 15: File not found (Ошибка 15: Файл не найден) и останавливается.

Как только ядро стартует, оно монтирует корневую файловую систему с жесткого диска. Имя раздела, содержащего эту файловую систему, передается ядру как параметр, который вы видели в файле **menu.lst** выше. Монтирование корневой файловой системы – ключевая точка процесса загрузки, и если вы пытаетесь точно определить проблему, жизненно важно выяснить, удалось ли ядру это сделать. Сбой в монтировании файловой системы, как правило, приводит к «панике ядра»

За ушко притянуто

Начальная загрузка – это как ушко на голенище сапога для верховой езды, которое помогает вам его надеть. Концепция «вытяни себя сам за ушко» означает поднять себя над землей, потянув за него [барон Мюнхаузен таким образом вытащил себя за волосы из болота, – *прим. пер.*]. Это, конечно, невозможно, но идея такого «ушка» – «спасения утопающих руками самих утопающих» – работает. И она особенно подходит к задаче старта компьютера.

```
    [ Minimal BASH-like line editing is supported.   For
      the   first   word,  TAB  lists  possible  command
      completions.  Anywhere else TAB lists the possible
      completions of a device/filename. ]

grub> help
blocklist FILE                         boot
cat FILE                               chainloader [--force] FILE
clear                                  color NORMAL [HIGHLIGHT]
configfile FILE                        device DRIVE DEVICE
displayapm                             displaymem
find FILENAME                          geometry DRIVE [CYLINDER HEAD SECTOR [
halt [--no-apm]                        help [--all] [PATTERN ...]
hide PARTITION                         initrd FILE [ARG ...]
kernel [--no-mem-option] [--type=TYPE] makeactive
map TO_DRIVE FROM_DRIVE                md5crypt
module FILE [ARG ...]                  modulenounzip FILE [ARG ...]
pager [FLAG]                           partnew PART TYPE START LEN
parttype PART TYPE                     quit
reboot                                 root [DEVICE [HDBIAS]]
rootnoverify [DEVICE [HDBIAS]]         serial [--unit=UNIT] [--port=PORT] [--
setkey [TO_KEY FROM_KEY]               setup [--prefix=DIR] [--stage2=STAGE2_
terminal [--dumb] [--no-echo] [--no-ed terminfo [--name=NAME --cursor-address
testvbe MODE                           unhide PARTITION
uppermem KBYTES                        vbeprobe [MODE]

grub>
```

› **Рис. 2. Результат вывода команды "help", напечатанной в командной строке *Grub*.**

[kernel panic], хотя некоторые системы просто зависают. Если же ядро благополучно смонтировало корневую файловую систему, оно создает одиночный процесс (с ID 1), выполняющий программу **/sbin/init**. Если ядро не может найти init, оно также «паникует» и останавливается или (в зависимости от дистрибутива) выкидывается в командную строку суперпользователя. Да, кстати, придется добавить вам путаницы: Ubuntu больше не использует **init**, он заменен на **upstart**.

Init отвечает за запуск скриптов, которые запустят все остальные сервисы в системе. Существует очень важный и достаточно низкоуровневый скрипт, запускаемый **init** в начале процесса. В Red-Hat-подобных системах это **/etc/rc.d/rc.sysinit**, в SUSE – **/etc/init.d/boot**. Среди прочего, этот первичный скрипт последовательно проверяет и монтирует остальные разделы диска, прописанные в **/etc/fstab**. Хотя, конечно, на этом этапе тоже хватает потенциальных неприятностей, но рассказ о них придется оставить до следующего месяца.

Берем Grub под контроль

Ключевое умение в исправлении проблем загрузки – это знание, как вручную вмешаться в процесс загрузки *Grub*. Большинство дистрибутивов конфигурируют *Grub* на выбор пункта меню «по умолчанию», но на короткое время (пару секунд) отображают окно, где вы можете нажать Esc для прерывания процесса и принятия полного контроля над *Grub* на себя. Обычно это ведет к выходу из заставки *Grub* в текстовое меню. Здесь, следуя инструкциям на экране, выберите пункт меню и отредактируйте связанные с ним команды перед загрузкой. Можно даже переключиться в командную строку *Grub* и вводить команды *Grub* напрямую; например в этом месте вы можете, теоретически, вручную ввести root, kernel и initrd строки из файла **menu.lst**, который мы рассмотрели ранее. Рис. 2 показывает результат ввода help в командную строку *Grub*.

Спасательная загрузка

Если все способы «подкрутки» команд в меню загрузки *Grub* не помогли загрузить систему, может быть, настало время выполнить 'rescue boot' [спасательную загрузку], которая означает, что вы должны загру- »

» зить Linux с инсталляционного CD или другого «средства спасения». Ядро и его модули загружаются с CD вместе с небольшой файловой системой, хранящейся в памяти. В результате запускается маленький, но работоспособный Linux, не привязанный ни к какой файловой системе на жестком диске. Потом вы можете смонтировать разделы жесткого диска в файловую систему и получить к ним доступ для устранения неполадки. Помогут установочные CD или DVD большинства современных дистрибутивов, так как для восстановления определенного дистрибутива не требуется именно его установочный диск, и вы можете взять любой доступный.

Пользуясь спасательной системой, очень важно иметь ясную голову, потому что файлы на вашем жестком диске окажутся не в тех местах, в которых вы видели их при «реальной» установке. Например, если в спасательной системе смонтировать корневой раздел в **/mnt**, то файл, обычно видимый в **/etc/fstab**, нужно будет искать в **/mnt/etc/fstab**.

Скорая помощь

Хотите узнать больше о лабиринтах Grub? Посетите сайт проекта GNU для просмотра и/или загрузки GNU Grub Manual по адресу **www.gnu.org/software/Grub/manual/**. Он доступен в различных форматах, что позволит работать с ним на вашей привычной системе.

Реальная ситуация 1

Наш первый пример касается RHEL5, в котором директория **/usr** помещена в отдельный раздел. По какой-то причине (забыл, какой) мне пришлось диктовать по телефону, как редактировать **/etc/fstab**, администратору-новичку, и в строке, отвечающей за монтирование раздела **/usr**, он закончил поле словами LABEL=/user вместо LABEL=/usr. Эта ошибка приводит к невозможности многопользовательской работы, выкидывая вас в однопользовательскую оболочку. Диалог выглядит примерно так:

```
fsck.ext3: Unable to resolve 'LABEL=/user'          [FAILED]
*** An error occurred during the file system check.
*** Dropping you to a shell; the system will reboot
*** when you leave the shell
Give root password for maintenance (or type Control-D to continue):
```

Тут главное – действительно прочитать сообщение об ошибке. Почему система пытается найти метку с именем /user? В каком файле это происходит?

Хм... Может быть, **/etc/fstab**? Так как система любезно вытолкала нас в командную строку root, вы решаете, что можно непосредственно отредактировать **/etc/fstab** и исправить ошибку. Однако оказывается, что на данном этапе (по милости fsck) корневая файловая система монтируется в режиме «только для чтения», и вы не можете редактировать файлы и записывать изменения. Здесь сработает следующий трюк. Перемонтируем файловую систему в режиме чтения-записи, вот так:

```
# mount -o remount,rw /
```

Выполнив это, я смог отредактировать **/etc/fstab**, исправить ошибку и нормально перезагрузиться.

Реальная ситуация 2

Второй пример сложнее предыдущего, и включает сценарий двойной загрузки.

Сперва я установил OpenSUSE 10.2 на пустой жесткий диск, выделив 8-ГБ раздел **hda1** под корневой и 2-ГБ раздел **hda2** под раздел подкачки. Установщик немедля прописал загрузчик *Grub* 1-го этапа в основную загрузочную запись, а загрузчик полуторного этапа – в блоки диска. Немного погодя я установил на свободное место диска Fedora 7, выделив под корневой раздел **hda3** те же 8 ГБ и разделив swap-раздел OpenSUSE **hda2** между двумя системами. **Рис. 3** вверху на следующей странице поможет вам, если вы уже запутались. Я разрешил Fedora установку *Grub* по умолчанию в **/dev/sda** (имена отличаются, так как OpenSUSE 10.2 называет разделы традиционно – **hda1**, **hda2** и т.д., а Fedora 7 использует SCSI-стиль именования – **sda1**, **sda2** и т.д.).

При установке Fedora мне было предложено добавить дополнительный пункт в загрузочное меню *Grub*, и я попросил добавить пункт с названием SUSE 10.2 и именем устройства **/dev/sda1**. В результате файл конфигурации *Grub* у Fedora выглядел так:

```
title SUSE 10.2
    rootnoverify (hd0,0)
    chainloader +1
```

Если я выбираю этот пункт меню, *Grub* Fedora должен начать загрузку *Grub* OpenSUSE с первого сектора первого раздела (который *Grub* называет (hd0,0)), Fedora7 – **/dev/sda1**, а OpenSUSE **/dev/hda1** – ну надо ли так морочить людям голову?). Итак, теперь я мог выбрать между Fedora и SUSE в загрузочном меню, хотя несколько нудно, что SUSE при загрузке опять выводит меню *Grub* и требует выбрать себя в нем. Будь я умнее, я бы скопировал строчку, загружающую SUSE из файла конфигурации *Grub* с **hda1** в файл на **hda3**, что по идее удобнее, так как можно загружаться прямо в SUSE, минуя второе меню.

Выкрутасы Fedora

Так или иначе, через пару недель я понял, что Fedora мне ни к чему, и решил переразметить ее раздел из SUSE. И вот (загрузившись в SUSE), я запустил:

```
# mke2fs -j /dev/hda3
```

для создания новой файловой системы на разделе. Напомню, что **hda3** был корневым разделом Fedora. Затем я смонтировал заново созданный раздел в **/mnt**. Все прошло хорошо, и я перезагрузился. Тут-то и вскрылась язва, потому что в этом месте *Grub* отказал. Я не мог загрузиться в Fedora – чего и следовало ожидать, раз я удалил ее раздел, но, что более серьезно, SUSE тоже не запускалась. Все, что происходило – появление слова "GRUB" на экране. В этом пункте секрет вашего поведения таков: Без паники! Прокрутите события назад. Снова взгляните на **Рис. 3**, и вспомните, что содержимое **hda3** теперь перезаписано. Загрузчик этапа 1.5, установленный Fedora, ссылается на загрузчик этапа 2 в файловой системе **hda3**, которого не существует. Все, что нам надо – это переустановить исходные файлы *Grub* этапов 1 и 1.5 для восстановления возможности загрузки SUSE 10.2.

Единственный способ их вернуть – загрузиться со спасательного диска. Как я уже отметил, для этого годится любой инсталляционный CD Linux, но я выбрал Ubuntu 7.04, поскольку это и Live CD, и ОС Linux, включая рабочий стол Gnome, работающий прямо с CD.

Ubuntu, как и Fedora, называет разделы диска как **sda1**, **sda2**, поэтому, как только мой Live CD загрузился, я смог примонтировать корневой раздел SUSE в файловую систему Ubuntu:

```
$ sudo mount /dev/sda1 /mnt
```

(Если вы недоумеваете, зачем в этой команде sudo, поясню, что в Ubuntu запрещен прямой вход как суперпользователю, и при каждом запуске команды с привилегиями root требуется использовать *sudo*. На Live CD пароль для этого не нужен.) Теперь мне нужно запустить *Grub* и перезаписать MBR. На этом этапе неясно, где находится команда grub (мне нужна копия для SUSE, то есть та, что на **hda1**), но запустив

```
$ sudo find /mnt –name grub
```

я быстро обнаружил, что это **/mnt/usr/sbin/grub**. Теперь я смог запустить его и ввести команды в командной строке *Grub*, как показано здесь:

```
$ sudo /mnt/usr/sbin/grub
grub> root (hd0,0)
grub> setup (hd0) (hd0,0)
 Checking if "/boot/grub/stage1" exists... yes
```

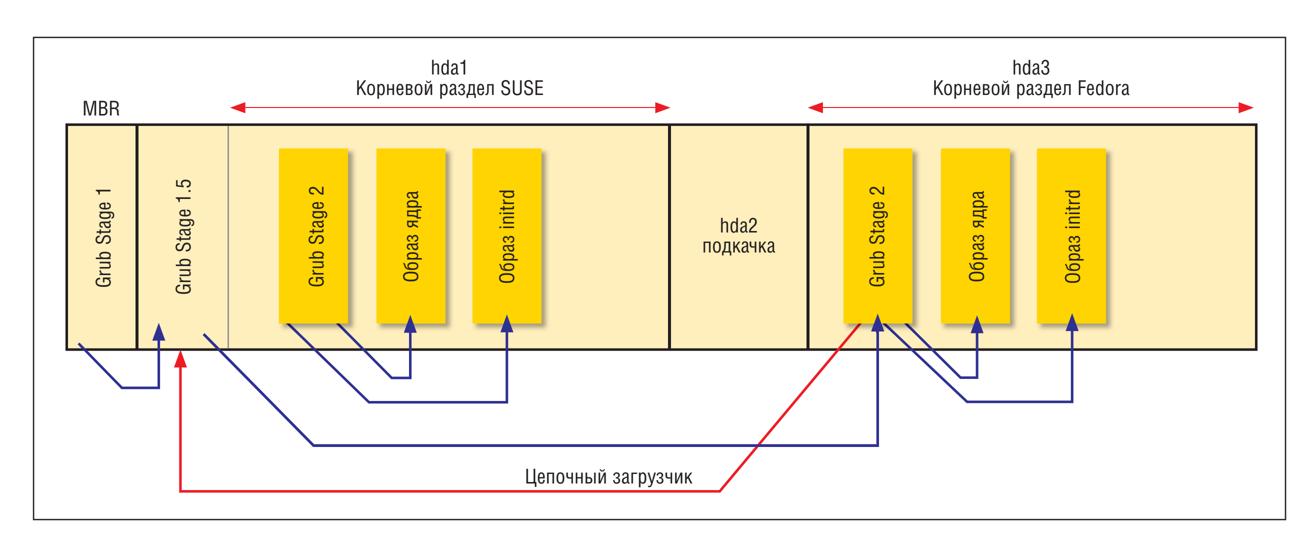


Рис. 3. Конфигурация с двойной загрузкой: взаимодействие между разделами SUSE и Fedora.

Параметры ядра

Частая причина модификации команд *Grub* – передача загрузочных параметров ядру. Эти параметры просто добавляются в конец строки kernel. Список загрузочных параметров ядра покажет команда man bootparam, или можете купить книгу Грега Кроа-Хартмана [Greg Kroah-Hartmann] «Linux Kernel in a Nutshell» (а где еще найдешь ядро? Прочитайте ее бесплатно на www.kroah.com/lkn/), там про это целая глава. Если у вас установлены исходные коды ядра, вы найдете его параметры в **/usr/src/linux/Documentation/kernel-parameters.txt**. Помните, что многие параметры являются необязательными, и могут или не могут быть настроены в ядре. В таблице они приведены лишь для того, чтобы предоставить вам идею. Запустив Linux, вы можете проверить загрузочные параметры, которые были переданы в него, изучив **/proc/cmdline**.

```
Checking if "/boot/grub/stage2" exists... yes
Checking if "/boot/grub/e2fs_stage1_5" exists... yes
Running "embed /boot/grub/e2fs_stage1_5 (hd0)"... 15 sectors
are embedded. succeeded
Running "install /boot/grub/stage1 (hd0) (hd0)1+15 p (hd0,0)/
boot/Grub/stage2 /boot/grub/menu.lst"... succeeded
Done.
grub> quit
```

Ключевая команда здесь setup (hd0) (hd0,0), которая велит *Grub* переустановить оригинальный (SUSE) stage 1 в MBR на **(hd0)** и скопировать исходный (SUSE) stage 1.5 в сектора, следующие за ним (для более детального описания этого смотрите команды install и embed в руководстве GNU на www.gnu.org/software/Grub/manual/Grub.html). Проделав это, я смог просто перезагрузиться и... О-ля-ля! Мое загрузочное меню *Grub* SUSE вернулось!

Международные спасатели

Live CD вроде Ubuntu или Knoppix прекрасно делают свою работу по спасению загрузки, но кроме них, существуют более маленькие и быстрые диски для восстановления. Некоторые из вас заинтересуются SystemRescueCD (см. www.sysresccd.org) – он не только загружается гораздо быстрее, но также содержит расширенную коллекцию инструментов для управления и редактирования разделов, включая и классику вида *gparted*, и более экзотические инструменты, вроде NTFS-3g, позволяющий получить доступ к разделам Windows NTFS с полной поддержкой чтения и записи – великолепно для восстановления файлов на старых вышедших из строя разделах Windows XP! Другая вещь, выгодно отличающая этот восстановительный CD – он может кэшировать свою файловую систему в памяти, позволяя вам при необходимости вынуть CD и вставить другой. LXF

Команды ядра

Параметр	Действие
quiet	Подавляет ведение журнала ядра, за исключением предупреждений и ошибок. Не используйте его, если вы пытаетесь решить проблему с загрузкой.
splash	Выводит заставку и прогресс-индикатор, закрывающие сообщения ядра. Удалив ее, вы можете отследить ошибку.
single or S	Велит ядру при запуске **init** заставлять систему запускаться в однопользовательском уровне выполнения [run level]. В зависимости от конфигурации у вас спросят или не спросят пароль root; затем вас пустят в оболочку root. X-сервер или графический рабочий стол не стартуют, никакие разделы не монтируются (конечно, за исключением корневого). Это действенный режим для отслеживания проблем X или восстановления файловой системы.
vga=ask	Эта опция позволяет вам выбрать режим видеоадаптера. Видеорежим устанавливает разрешение экрана (в виде строк и колонок), и добавив эту опцию, вы сможете быстро выбрать видеорежим из списка. Полезна, если ваш монитор не поддерживает режим по умолчанию.
acpi=off	Отключает ACPI (Advanced Configuration and Power Interface).
init=/bin/sh	Запускает программу **/bin/sh** (оболочку) взамен **init**. Это даже более экстремально, чем выполнять однопользовательскую загрузку и фактически представляет режим пользователя без инициализации. Загрузка будет исключительно быстрой, и оболочка будет единственным запущенным процессом пользовательского пространства.

Диски – долой! Научите ПК с Linux загружаться через локальную сеть

Сеть: Загрузка

Хотите превратить свой ПК в домашний кинотеатр или завести кластер из однотипных компьютеров? Ответ – сетевая загрузка. Нейл Ботвик все объяснит.





Наш эксперт

Нейл Ботвик считает, что Gentoo – лучший дистрибутив после его предыдущего любимца, и никогда не упускает возможности исправить то, что не было сломано.

Помните времена, когда компьютерам не были нужны жесткие диски? Amiga и Atari ST загружали все необходимое с дискеты. А еще раньше у 8-битных компьютеров типа Commodore 64 и Spectrum даже и дискет-то не было, все необходимое загружалось из ПЗУ. Сегодня сложно купить компьютер без жесткого диска в несколько сотен гигабайт, но зато загрузить компьютер можно вообще без винчестера. Так называемая «загрузка по сети» позволяет такому компьютеру получить все необходимое с другого компьютера через сеть.

Это рождает два правомерных вопроса: зачем и как. Зачем бы нам это делать? Причин несколько: может быть, вам нужен тихий и прохладный компьютер, а жесткие диски нагреваются и изрядно шумят. Например, в моей гостиной стоит компьютер для работы MythTV, с материнской платой Via EPIA без кулеров и без жесткого диска, абсолютно бесшумный. Объединив все жесткие диски в один большой диск или массив дисков, вместо использования отдельных дисков на каждом компьютере, можно более эффективно обслуживать кластеры. То же справедливо и для компьютерной сети в интернет-кафе или

> «У моего MythTV-ПК в гостиной нет винчестера, и он абсолютно тихий.»

в школьном кабинете информатики. Это не то же самое, что использовать тонкого клиента, когда всю работу выполняет сервер. Компьютер, загружаемый через сеть – это обычный компьютер или рабочая станция; вся разница – что его жесткий диск подключен длинным сетевым кабелем вместо короткого шлейфа.

На второй вопрос (как это сделать?) мы ответим дальше. Как и на первый вопрос, на второй есть несколько ответов, но мы ограничимся только одним. PXE (Preboot eXecution Environment или pixie – среда выполнения перед загрузкой) – самый простой способ обеспечить запуск через сеть, если клиентский компьютер (без жесткого диска) его поддерживает. На большинстве современных материнских плат есть по меньшей мере одна сетевая карта с поддержкой PXE, так что это самый простой вариант; прочие устройства могут потребовать другого подхода, например, установки ПЗУ etherboot (www.etherboot.org) на сетевой карте или инициализации процесса с дискеты или CD. Но мы их здесь не рассматриваем.

Как это работает?

Перед тем, как рассматривать загрузку с сети, вспомним, как происходит обычная загрузка с жесткого диска:

1. BIOS компьютера ищет код загрузчика ОС (bootloader) в первом секторе жесткого диска и загружает его.
2. Загрузчик ОС, используя информацию из файла настройки и пользовательского меню, загружает ядро, передавая ему информацию о местонахождении корневого раздела и другие данные.
3. Ядро монтирует корневой раздел и запускает процесс /sbin/init, который загружает все остальное.

Загрузка с использованием PXE очень похожа на обычную загрузку:

Системные требования

Клиент
PXE-совместимая сетевая карта. Если у вас встроенная сетевая карта, проверьте, есть ли в меню или настройках BIOS возможность загрузки из сети.

Много оперативной памяти
Подкачка через NFS работает крайне медленно, не говоря уже о ее ненадежности. Поэтому для запуска приложений требуется физическое ОЗУ приличного объема.

Большой жесткий диск
В зависимости от того, сколько программ установлено, потребуется от 2 до 5 ГБ на каждого клиента, плюс место на домашние каталоги и файлы данных.

Клиент и сервер
Необходимо быстрое соединение, особенно если клиентов несколько. Операции, интенсивно использующие обращения к диску, ощутимо замедлятся, поэтому неплохо иметь соединение в 1 Гбит/с. Если интенсивных обращений к диску нет, скорость может быть гораздо ниже. Мой клиент MythTV использует Powerline-соединение [через бытовую электросеть, – *прим.ред.*], скоростью около 30 Мбит/с. Загружается компьютер немного медленнее обычного, но все остальное работает вполне нормально.

без винчестера

1 Код PXE в сетевой карте или в BIOS компьютера рассылает по сети расширенный DHCP-запрос.
2 Должным образом сконфигурированный сервер возвращает информацию о настройках DHCP, а также адрес сервера загрузки, код сетевого загрузчика и загрузочное меню, подобное обычному меню загрузки Grub или Lilo.
3 Когда пользователь выбирает пункт меню (или истекает время ожидания), загрузчик PXE загружает ядро с указанного сервера загрузки, определяя расположение корневого раздела и другие параметры.
4 Ядро монтирует корневой раздел с помощью NFS и запускает процесс **/sbin/init**, который загружает все остальное.

Во всем этом участвуют три сервера: сервер DHCP, отвечающий на начальный запрос, возвращая информацию о настройках, сервер загрузки, содержащий ядро, и сервер NFS, содержащий корневую файловую систему, которая будет монтироваться при загрузке компьютера клиента. Они могут располагаться как на одном компьютере, так и на разных – выбор за вами – хотя проще располагать их вместе, особенно сервер DHCP и серверы загрузки.

Предупреждение о перегрузке аббревиатурами: здесь вы встретите множество ТБС и РТБС. Мы уже использовали PXE, BIOS, DHCP и NFS, а будут и новые. Запомните их, так как чуть позже вас ждут тесты.

Установка системы

Во-первых, нам потребуется установленная система на сервере, с которого мы будем загружаться. Использовать ту систему, которая на нем уже стоит, нельзя: даже если аппаратная часть компьютеров полностью идентична, некоторые детали (особенно настройки системы в **/etc**) могут различаться. Для каждого клиента нужно установить ОС в отдельный каталог; обычно это **/diskless/IPaddress**, что позволяет вам хранить все установленные ОС клиентов в одном месте и без конфликтов, потому что два компьютера не могут иметь одинаковые IP-адреса.

Простейший способ выполнить начальную установку системы – это подключить к будущему бездисковому компьютеру винчестер и привод CD/DVD и установить систему обычным образом. Если вы будете настраивать загрузку по сети в существующей системе, то этот этап уже пройден. Потом содержимое диска можно скопировать в соответствующий каталог на сервере – либо физически подключив диск, либо скопировав файлы через сеть. Поскольку вам когда-нибудь придется открывать сетевой доступ к каталогу сервера, сделаем это сейчас и скопируем файлы. В этом примере адрес сервера 192.168.2.10, а адрес клиента 192.168.2.50. Все команды необходимо выполнять с привилегиями администратора.

На сервере добавьте строки

```
/diskless/192.168.2.50  192.168.2.0/255.255.255.0(rw,sync,no_root_squash,no_all_squash,no_subtree_check,rw)
```

в файл **/etc/exports**, запустите

```
mkdir -p /diskless/192.168.2.50
exportfs -a
exportfs
```

и убедитесь, что в выводе последней команды есть новый каталог. На клиенте запустите

```
mkdir -p /mnt/remoteroot
mount -t nfs /mnt/remoteroot 192.168.2.10:/diskless/192.168.2.50
```

```
Shell - Konsole
Session Edit View Bookmarks Settings Help
[root@desiato /diskless 0]% tail -f /var/log/dnsmasq.log
Oct 12 08:48:24 dnsmasq[4975]: DHCPREQUEST(eth0) 192.168.1.14 00:40:63:ee:9c:0a
Oct 12 08:48:24 dnsmasq[4975]: DHCPACK(eth0) 192.168.1.14 00:40:63:ee:9c:0a fook
Oct 12 09:15:17 dnsmasq[4975]: DHCPREQUEST(eth0) 192.168.1.155 00:0c:29:aa:6a:85
Oct 12 09:15:17 dnsmasq[4975]: DHCPACK(eth0) 192.168.1.155 00:0c:29:aa:6a:85 Knoppix
Oct 12 09:15:33 dnsmasq[4975]: DHCPREQUEST(eth0) 192.168.1.155 00:0c:29:aa:6a:85
Oct 12 09:15:33 dnsmasq[4975]: DHCPACK(eth0) 192.168.1.155 00:0c:29:aa:6a:85 Knoppix
Oct 12 10:07:48 dnsmasq[4975]: DHCPREQUEST(eth0) 192.168.1.155 00:0c:29:aa:6a:85
Oct 12 10:07:48 dnsmasq[4975]: DHCPACK(eth0) 192.168.1.155 00:0c:29:aa:6a:85 Knoppix
Oct 12 10:08:04 dnsmasq[4975]: DHCPREQUEST(eth0) 192.168.1.155 00:0c:29:aa:6a:85
Oct 12 10:08:04 dnsmasq[4975]: DHCPACK(eth0) 192.168.1.155 00:0c:29:aa:6a:85 Knoppix


Oct 12 10:50:22 dnsmasq[4975]: DHCPDISCOVER(eth0) 00:40:63:ee:9c:0a
Oct 12 10:50:22 dnsmasq[4975]: DHCPOFFER(eth0) 192.168.1.14 00:40:63:ee:9c:0a
Oct 12 10:50:24 dnsmasq[4975]: DHCPREQUEST(eth0) 192.168.1.14 00:40:63:ee:9c:0a
Oct 12 10:50:24 dnsmasq[4975]: DHCPACK(eth0) 192.168.1.14 00:40:63:ee:9c:0a fook
Oct 12 10:50:24 dnsmasq[4975]: TFTP sent /diskless/192.168.1.14/boot/pxegrub to 192.168.1.14
Oct 12 10:50:24 dnsmasq[4975]: TFTP error 0 TFTP Aborted received from 192.168.1.14
Oct 12 10:50:24 dnsmasq[4975]: TFTP failed sending /diskless/192.168.1.14/boot/pxegrub to 192.168.1.14
Oct 12 10:50:24 dnsmasq[4975]: TFTP sent /diskless/192.168.1.14/boot/pxegrub to 192.168.1.14
Oct 12 10:50:35 dnsmasq[4975]: DHCPDISCOVER(eth0) 00:40:63:ee:9c:0a
Oct 12 10:50:35 dnsmasq[4975]: DHCPOFFER(eth0) 192.168.1.14 00:40:63:ee:9c:0a
Oct 12 10:50:35 dnsmasq[4975]: DHCPREQUEST(eth0) 192.168.1.14 00:40:63:ee:9c:0a
Oct 12 10:50:35 dnsmasq[4975]: DHCPACK(eth0) 192.168.1.14 00:40:63:ee:9c:0a fook
Oct 12 10:50:35 dnsmasq[4975]: TFTP sent /diskless/192.168.1.14/boot/grub.lst to 192.168.1.14
Oct 12 10:50:41 dnsmasq[4975]: TFTP sent /diskless/192.168.1.14/boot/vmlinuz to 192.168.1.14
Oct 12 10:50:53 dnsmasq[4975]: DHCPDISCOVER(eth0) 00:40:63:ee:9c:0a
Oct 12 10:50:53 dnsmasq[4975]: DHCPOFFER(eth0) 192.168.1.14 00:40:63:ee:9c:0a
Oct 12 10:50:53 dnsmasq[4975]: DHCPREQUEST(eth0) 192.168.1.14 00:40:63:ee:9c:0a
Oct 12 10:50:53 dnsmasq[4975]: DHCPACK(eth0) 192.168.1.14 00:40:63:ee:9c:0a fook

[root@desiato /diskless 130]%
Shell  Shell No. 2
```

› **Хороший способ наблюдать за загрузкой в реальном времени - просматривать содержимое файлов журналов с помощью tail -f.**

```
rsync -ax / /mnt/remoteroot
rsync -ax /home/ /mnt/remoteroot/home/
rsync -ax /boot/ /mnt/remoteroot/boot/
```

Две последних строки в этом примере нужны только в том случае, когда домашний каталог (home) и каталог загрузки (boot) находятся в различных разделах. Проще установить все в один раздел.

А что делать, если жесткий диск к клиенту подключить нельзя? Тогда установите систему на сервер в раздел, который позже смонтируете как **/diskless**. Проверьте, чтобы все настройки, задаваемые в процессе этой установки, были совместимы с аппаратной частью клиентской машины. Не устанавливайте загрузчик, и после перезагрузки сервер загрузится, как обычно. Теперь смонтируйте новый раздел как **/diskless**, создайте в нем каталог **IPaddress** и переместите остальное содержимое в этот каталог. Пользователям Gentoo сделать это проще, так как они могут установить систему вручную в каталог **/diskless/IPaddress** (просто используйте этот каталог при обычной установке вместо **/mnt/gentoo**).

> **«Диск не подключить к клиенту? Убедитесь, что настройки подходят к его аппаратной части.»**

Удаленная установка системы отличается от обычной установки на локальный диск. Прежде всего, может потребоваться перекомпиляция ядра. На это есть две причины. Первая – ради возможности смонтировать корневой раздел по сети вы должны установить опцию ROOT_NFS=y. Вторая – в большинстве дистрибутивов для загрузки модулей, необходимых для работы оборудования, используется *initrd* (началь- »

» ный ram-диск). Хотя к нему можно обращаться через сеть, скорее всего его придется модифицировать, а это посложнее перекомпиляции ядра. Механизмы перекомпиляции ядра выходят за рамки этой статьи, но вы можете найти руководство в LXF89. Если этого номера у вас нет, возьмите статью в формате PDF на стороне 1 DVD, в разделе **Журнал/PDFs**. Пользователи SUSE (по крайней мере 10.2) могут расслабиться – этот дистрибутив уже подготовлен для загрузки по сети с использованием PXE. Если команда

```
zgrep ROOT_NFS /proc/config.gz
```

покажет ROOT_NFS=y, ваше ядро готово к сетевой загрузке и следующий абзац можно пропустить.

Непременно вкомпилируйте сетевой драйвер в ядро – не оставляя его отдельным модулем; то же лучше сделать и с остальными драйверами. lspci поможет определить используемое аппаратное обеспечение, после этого система справки ядра должна вывести список необходимых драйверов. В идеальном мире все нужные модули должны вписаться в ядро, а все ненужные – исчезнуть. В жизни так бывает далеко не всегда. В частности, у пользователей часто появляются проблемы после компиляции в ядро драйверов ALSA. Если вы устанавливаете систему на жесткий диск, подключенный к клиенту описанным выше способом, то перед переносом чего-либо на другой компьютер перекомпилируйте ядро – возможно, потребуется несколько попыток, чтобы все получилось.

Последняя задача – отредактировать **/etc/fstab**. Удалите оттуда все строки, относящиеся к swap-файлу – использовать его в NFS не рекомендуется – и измените строку для корневого раздела на

```
192.168.2.10:/diskless/192.168.2.50 / nfs defaults,nolock 0 0
```

Скорая помощь

По возможности, проверяйте все локально. Перед тем, как переносить систему на сервер, убедитесь, что она загружается с жесткого диска. Это повысит шансы на успешный результат.

DHCP, TFTP и другие аббревиатуры

Теперь, когда система на сервере полностью установлена, нужно объяснить серверу, как перенести ее на компьютер клиента. Для этого нам понадобятся DHCP-сервер и TFTP-сервер. DHCP (Dynamic Host Configuration Protocol – протокол динамической конфигурации хоста) – это система, с помощью которой компьютер может запросить IP-адрес и другие параметры конфигурации непосредственно из сети. Помните, что на клиенте нет постоянного хранилища файлов, поэтому хранить статические настройки сети негде. TFTP (Trivial File Transfer Protocol – простой протокол передачи файлов) – это простая система передачи файлов, удобная для сетевой загрузки, потому что ее можно хранить в ПЗУ сетевой карты PXE. Она используется для передачи загрузчика ОС и ядра на компьютер клиента.

Серверов DHCP и TFTP на свете хватает; *dnsmasq* (**www.thekelleys.org.uk/dnsmasq**) является легко настраиваемой комбинацией обоих. Установка *dnsmasq* по умолчанию предоставляет базовый сервис DHCP, но нам потребуется еще кое-что. Установив и запустив *dnsmasq*, отключите сервер DHCP в своем маршрутизаторе, если он у вас есть, иначе может возникнуть конфликт. *Dnsmasq* будет наблюдать за всей сетью, а не только за компьютерами без жесткого диска.

Сначала нужно позаботиться, чтобы dnsmasq всегда выделял клиенту один и тот же IP-адрес. Для этого нужен MAC-адрес клиента (еще одна аббревиатура). Чтобы получить MAC-адрес, запустите на этом компьютере команду ifconfig или загрузитесь с подключенного диска или live CD – MAC-адрес зашит в сетевой карте и не может меняться [это не совсем так: существуют программы, способные менять MAC-адрес карты, но в условиях данного урока о них можно забыть, – *прим. ред.*]. Раскомментируйте строку conf-dir=/etc/dnsmasq.d в файле **/etc/dnsmasq.conf**, затем создайте файл **/etc/dnsmasq.d/local** и добавьте в него такую строку:

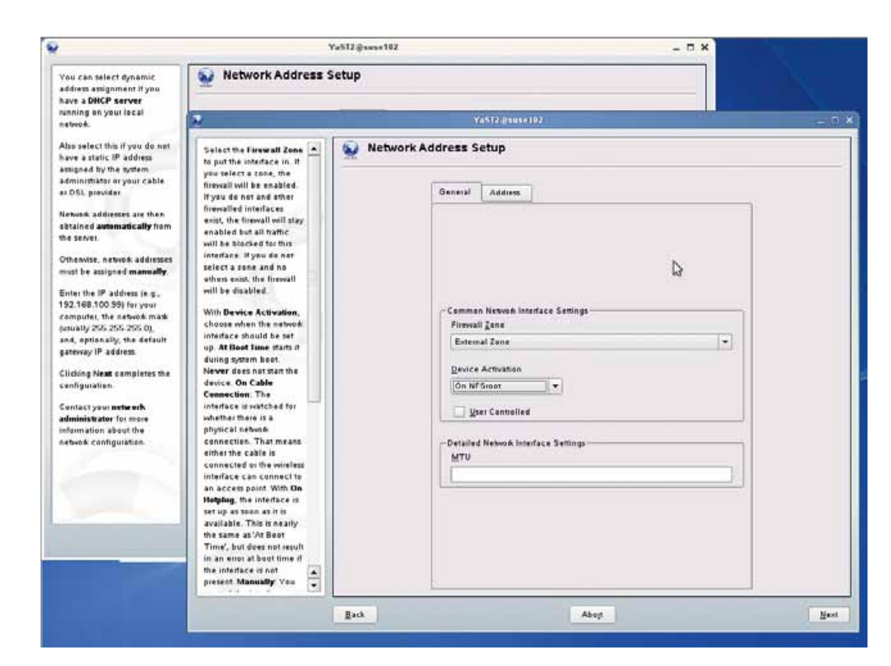
```
dhcp-host=00:0c:29:d4:cb:2e,192.168.2.50,somename
```

где 00:0c:29:d4:cb:2e – MAC-адрес. Эта строка означает, что компьютеру с данным MAC-адресом всегда будет выделяться IP-адрес 192.168.2.50 и имя узла somename. То же можно проделать и для других компьютеров сети, чтобы им всегда выделялись одни и те же IP-адреса. Прямого отношения к бездисковой загрузке это не имеет.

Чтобы разрешить загрузку с *pxegrub*, добавьте в файл строки

```
log-facility=/var/log/dnsmasq.log
enable-tftp
tftp-root=/diskless
tftp-unique-root
dhcp-boot=/boot/pxegrub
dhcp-option=150,/boot/grub.lst
```

Первые три строки пояснять не нужно. Четвертая говорит dnsmasq о том, что для получения корневого каталога клиента нужно добавить его IP-адрес к каталогу tftp-root. Благодаря этому корневой каталог будет уникален для каждого клиента. Следующая строка определяет расположение файла загрузчика, который отправится клиенту, и последняя – меню загрузчика. Так и слышу ваш вопрос: «Какого загрузчика и меню?». Ну, мы еще не установили их, но сейчас это сделаем.



› SUSE 10.2 поддерживает загрузку по сети: включите соответствующую опцию в *YaST*.

Chroot'инг

А что делать, если в системе необходимо произвести какие-то изменения, например, обновить ПО на клиентских компьютерах? Можно зайти на каждый компьютер клиента и установить новое ПО, но клиент может быть просто выключен, когда вы захотите это сделать. Альтернативный вариант – произвести изменения с сервера с помощью chroot. Выполнив команды

```
mount -t proc none /diskless/IPaddress/proc
mount --bind /dev/ /diskless/IPaddress/dev
chroot /diskless/IPaddress /bin/bash
mount -a
source /etc/profile
```

вы окажетесь в командной оболочке системы клиента. Там можно сделать все необходимое, затем выйти:

```
umount -a
exit
umount /diskless/IPaddress/proc
umount /diskless/IPaddress/dev
```

Чтобы каждый раз не набирать эти команды снова, сохраните их как скрипт.

Добавляем сетевой загрузчик

Существует два основных загрузчика PXE – *pxegrub* and *pxelinux*. Мы возьмем *pxegrub*, но не потому, что *pxelinux* плох: просто *Grub* используется в большинстве дистрибутивов, форматы файлов конфигурации совместимы, и не нужно их изучать. *Grub*, способный на загрузку с

жесткого диска, CD/DVD и из сети, сейчас становится повсеместным. Если с *pxegrub* что-то не получится или вы желаете не быть как все, попробуйте *pxelinux*. Загрузчик входит в состав пакета *syslinux* (http://syslinux.zytor.com).

Pxegrub может быть уже установлен в вашей системе как часть пакета *Grub*, в SUSE он находится в **/usr/lib/grub/pxegrub**. *Dnsmasq* добавляет к этому пути корневой каталог клиента, и символьная ссылка на загрузчик будет такой: **/diskless/192.168.2.50/boot/pxegrub**. Можно скопировать файл загрузчика в этот каталог, но символьная ссылка всегда удобнее: тогда при установке новых версий *Grub* не придется переделывать загрузочные каталоги.

Для *pxegrub* нужно создать файл меню. Если вы уже редактировали стандартный файл меню *Grub*, то поймете его синтаксис.

```
default 0
timeout 10
title=default
root (nd)
kernel /boot/vmlinuz ip=dhcp root=/dev/nfs nfsroot=192.168.2.10:/
diskless/192.168.2.50
```

Основное отличие – в команде root, где для загрузки вместо привычного hd(n,m) указано сетевое устройство, и параметрах ядра, задающих использование DHCP для получения IP-адреса и разделяемого каталога NFS в качестве корневого раздела. Если нужно указать другие параметры ядра (такие как vga или noapic), просто припишите их к строке настроек.

И это функционирует?

Пора посмотреть, все ли работает. Для этого включите компьютер клиента и выберите загрузку из сети в настройках BIOS. Должны появиться сообщения, отражающие состояние DHCP; для его отслеживания на сервере также полезно запустить

```
tail -f /var/log/dnsmasq.log
```

Если все в порядке, то через несколько секунд появится меню *Grub*. Это значит, что DHCP и TFTP работают, и первый тест вами пройден (а вы думали, что тест будет про аббревиатуры?).

По окончании тайм-аута *Grub*, или после нажатия Enter, должен запуститься сам сетевой загрузчик. Выглядит это так же, как и обычная загрузка с жесткого диска, разве что медленнее. Если система нормально загружалась на клиентской машине с жестким диском, то и сейчас все должно работать. Единственная проблема, которая иногда возникает – *init* может попытаться еще раз запустить сетевое соединение, разрывая первоначальное, и все остановится, потому что исчезнет корневая файловая система. Решение проблемы зависит от дистрибутива – в некоторых есть возможность не перезапускать сеть, если она уже работает: например, пользователи Gentoo могут добавить строку

```
config_eth0=( "noop" )
```

в файл **/etc/conf.d/net**. А можно просто отключить запуск **eth0** в утилите управления сервисами вашего дистрибутива. В сетевых настройках SUSE есть опция управления NFSroot, делающая взаимодействие с сервисами ненужным. Точно так же нужно убедиться в том, что сеть не отключается и во время выключения компьютера, иначе система вновь потеряет связь со своей корневой файловой системой. В разных дистрибутивах это делается по-разному. Один из вариантов – просто удалить соответствующую символьную ссылку **K** в каталоге **runlevel**. Это будет что-то вроде **/etc/rc5.d/K05network**. **K** означает, что этот скрипт запускается при остановке системы для завершения (Kill) сервисов. Пользователи Gentoo могут воспользоваться хитрым приемом для предотвращения отключения сети при отключении интерфейса, добавив в файл **/etc/conf.d/**net следующие строки

```
predown() {
return 1
}
```

А оно работало?

На данный момент система должна загрузиться. Если это не так, прежде всего проверьте, смонтировалась ли корневая файловая система. Поищите в системном файле журнала на сервере любые сообщения, связанные с NFS. Самый простой способ получить их – перед запуском компьютера клиента выполнить следующую команду:

```
tail -f /var/log/messages | grep mountd
```

Имя файла может различаться в разных дистрибутивах и зависит от используемого системного сервиса журнала. На компьютере клиента должно вывестись что-то типа

```
authenticated unmount request from 192.168.2.50:nnn for /
diskless/192.168.2.50 (/diskless/192.168.2.50)
```

Это значит, что корневая файловая система смонтирована правильно. Увидев сообщение об ошибке, проверьте синтаксис **/etc/exports** и сверьтесь с man. Проверьте, можете ли вы смонтировать этот ресурс с другого компьютера вашей сети.

> «Остаток процесса загрузки такой же, как с локального диска: проверьте настройки и ядро.»

Если корневая система смонтирована правильно, все остальное не должно отличаться от обычной загрузки, поэтому проверьте настройки и конфигурацию ядра. Убедитесь, что вы не забыли добавить в конфигурацию ядра важные драйверы. Однако если вы проверяли установку на локальном жестком диске и корневая файловая система смонтировалась правильно, все остальное должно «просто работать». Готово! Теперь у вас есть компьютер без жесткого диска с полной установкой Linux, который можно использовать точно так же, как «обычный». LXF

Скорая помощь

Уменьшите количество сервисов и других программ, запускаемых при старте системы. Это позволит значительно ускорить загрузку, поскольку сетевая загрузка с необходимостью медленнее обычной.

Общий доступ к каталогам

Если вы хотите запускать несколько компьютеров без диска с одного сервера, то представляете, какой объем дискового пространства понадобится. Несколько каталогов (особенно **/usr** и **/opt**) можно разделить между клиентскими компьютерами (прочие каталоги не столь велики, чтоб о них беспокоиться, и одновременно безопасны для общего доступа), что позволит сэкономить около 3 ГБ дисковой памяти для каждого клиента. Но это непростая задача, требующая знаний в области скриптов загрузки. Просто создать символическую ссылку с каталога usr одного клиента на другой нельзя, потому что ссылки вне общего доступа NFS не работают. То же относится к монтированию с опцией —bind. Можно куда-нибудь поместить общие каталоги **/usr** и **/opt** (например, в **/diskless/common**) и добавить их в файл **/etc/exports**. После этого можно добавить строки монтирования **/usr** и **/opt** в файл **/etc/fstab** каждого клиента.

Пока все выглядит просто, но есть большая загвоздка, о которой некоторые из вас уже догадались. Оставить эту задачку вам было бы жестоко: скрипты init монтируют общие сетевые ресурсы при загрузке значительно позднее, чем локальную файловую систему, так как обычно надо дожидаться активизации сетевого интерфейса. Поэтому вам нужно было начать с изменения скриптов инициализации в вашем дистрибутиве. Быть может, игра не стоит свеч: даже если у вас много клиентов, то и дискового пространства много.

Программирование:

ЧАСТЬ 1: Любите покодировать, но нет свежих идей? **Майк Сондерс** затевает серию публикаций о полезных мини-программах. Начнем с персонального IRC-бота…



Наш эксперт

Майк Сондерс поддерживает сайт *Linux Format* и программирует со времен ZX Spectrum. В данный момент цель его жизни – изучить все известные человечеству варианты ассемблера.

Добро пожаловать в новую серию публикаций о программировании. В трех статьях этого цикла мы покажем, как писать полезные программы, а затем расширять и улучшать их по мере роста вашего умения. Ключевое слово здесь – «программа»: мы не будем копаться ни в теории, ни в сугубо техническом мусоре; вместо этого рассмотрим классные штуки, которые можно вытворять в различных языках программирования. Ведь гораздо интереснее писать настоящую программу, чем тратить время на изучение нудных конструкций циклов! А если вы уже пробовали кодировать, проблем с пониманием у вас не возникнет.

Начнем с IRC-бота, написанного на Perl. Если вы интернетчик со стажем, то наверняка пользовались IRC; если нет – прочтите врезку «Стоп… а что такое IRC?» на следующей странице, там объясняются базовые понятия. Коротко говоря, IRC – это чат в реальном времени, часто используемый разработчиками открытого ПО для взаимодействия друг с другом. Он очень быстр, в нем легко разобраться, и – главное – можно создавать виртуальных участников чата.

Этим мы и займемся – напишем «робота», который присоединяется к чату и рассказывает о компьютере. Скажем, вы на работе или в отпуске далеко от дома и хотите наблюдать за домашней системой (или за сервером на площадке провайдера). Наш IRC-бот будет сидеть себе в чате и ждать, когда появитесь вы и спросите его о времени работы системы и ее загрузке.

Программ для мониторинга системы немеряно, но они страдают повышенной дотошностью и забивают ваш почтовый ящик письмами с лишними подробностями о работе компьютера. С нашим ботом все намного проще: если вы хотите узнать, сколько осталось свободного места на диске, или посмотреть, хватает ли оперативной памяти, просто заскочите в IRC и спросите у него (через приватные сообщения). А самое важное – мы ограничим доступ так, чтобы эти данные могли получить только вы!

Букварь на Perl

Для начала пройдемся по основам Perl. Вы, небось, подумали: «А с какой стати писать этого бота на Perl?» Хороший вопрос. Ведь Perl – это язык скриптов, изначально задуманный для обработки текста. Однако он также снабжен массой модулей расширения, один из которых существенно упрощает взаимодействие с серверами IRC.

Perl сейчас предустановлен почти в каждом дистрибутиве; если в вашей системе его не оказалось, он почти наверняка найдется в менеджере пакетов. Это интерпретируемый язык, поддерживающий объектно-ориентированный подход и динамическую типизацию (то есть не нужно ни заранее объявлять переменные, ни задавать их тип). Последующие фрагменты кода набирать не надо: просто уясните, как они работают. Вывод строки элементарен:

```
print "Привет!";
```

Все выражения завершаются точкой с запятой (;), в стиле языка C. Имена переменных начинаются со знака доллара:

```
$num = 25;
$string = "гамбургеров";
print "Я съел $num $string\n";
```

Как видите, Perl не задумывается о типах переменных – он определяет их на лету. Аналогично C, символ \n означат перевод строки. Массивы объявляются с символом @:

```
@distros = ("Ubuntu", "Fedora", "SUSE");
print $distros[0];
```

Этот код выводит на экран слово "Ubuntu". По принципу многих других языков программирования, первый элемент массива получает номер 0. Для массивов полезна команда shift – она удаляет первый элемент массива и помещает его в переменную:

```
@distros = ("Ubuntu", "Fedora", "SUSE");
$foo = shift(@distros);
print "$foo\n";
$foo = shift(@distros);
print "$foo\n";
```

Мы создали массив из трех строк, затем сдвинули (shift) его первый элемент в переменную $foo, вывели ее и сдвинули массив еще раз. Итак, сначала выводится строка "Ubuntu", за ней – "Fedora", и в итоге в массиве остался единственный элемент, "SUSE". Расширенная версия массива – это «словарь» (или «хэш»), объявление которого начинается со знака процента (%):

```
%weight = ("Mike" => 8, "Paul" => 12, "Nick" => 25);
```

Здесь список слов связывается со значениями. Например, вы хотите вывести вес (weight) записи Nick:

```
print $weight{"Nick"};
```

Выведется 25. if устроен точно так же, как в большинстве других языков:

Месяц назад В последней статье про Mono Пол Хадсон решал общие проблемы.

личный IRC-бот



```
$x = 64;
if ($x == 64) {
print "Yep, x is 64!\n";
}
```

И, наконец, подпрограммы (функции) объявляются просто командой sub:

```
sub saystuff {
print "This is saystuff!\n";
}
saystuff();
```

Таковы «колесики и винтики» программирования на Perl – все, что нам нужно, чтобы начать писать IRC-бота!

Установка

Воспользуемся специальным модулем расширения языка Perl *Net::IRC*, предоставляющим интерфейс для протокола IRC – тогда нам не придется писать тонны кода для работы с сокетами и соединениями. Вместо этого мы попросим *Net::IRC* позаботиться о сетевых взаимодействиях, а сами будем дрессировать бота: пусть делает то, что нам нужно.

Вы можете найти *Net::IRC* с помощью своего менеджера пакетов (он называется **perl-irc** или **net.pm** или как-то вроде этого). А не найдете, не беда: собрать его из исходных текстов проще простого [даже термин «собрать» здесь избыточен: Net::IRC написан на Perl, так что речь идет просто о копировании его в нужный каталог, – *прим.ред.*]. Зайдите на сайт **http://search.cpan.org/dist/Net-IRC/** или откройте раздел **Development/Net-IRC** нашего **DVD** (первая сторона) и скопируйте файл **Net-IRC-0.75.tar.gz**.

Распакуйте архив, зайдите в образовавшийся каталог и введите команды:

```
perl Makefile.PL
make
make install
```

Последнюю из них нужно выполнять с правами администратора. Она установит модуль *Net::IRC*, и программа, которую мы напишем, сможет его применить!

Далее создадим две учетных записи для IRC: первая – для вас, вторая – для бота (если вы завсегдатай IRC, то первая у вас уже есть). Это важный этап, поскольку из-за борьбы со спамом на многих IRC-серверах пользователь должен зарегистрироваться, чтобы иметь возможность отправлять приватные сообщения другим участникам чата.

Зайдите в IRC и переключитесь в вашего обычного пользователя. Если вы еще не зарегистрированы, введите такую команду:

```
/msg nickserv register <password>
```

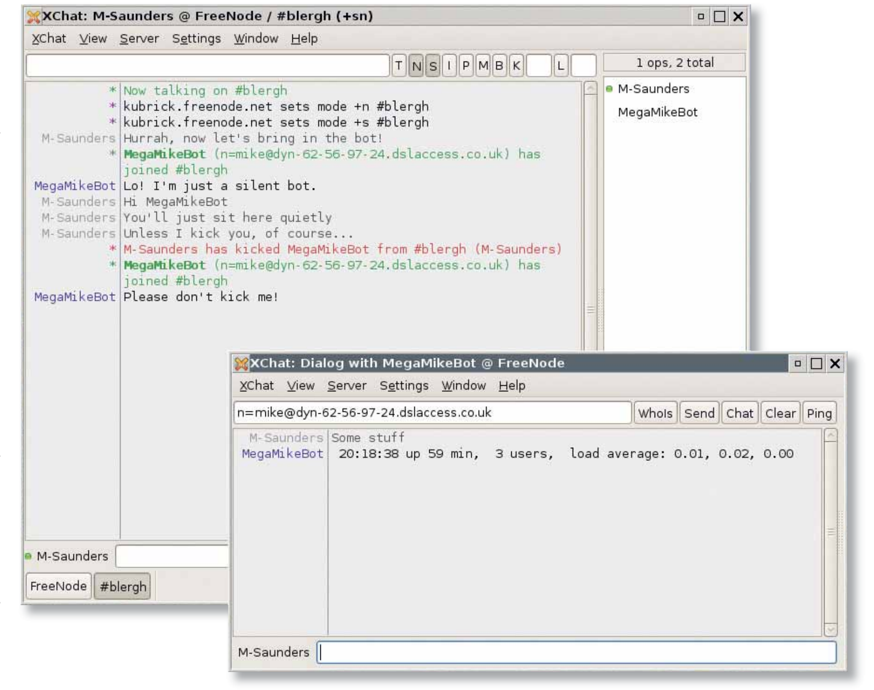
(<password> замените на подходящий пароль). Примеры команд приведены для серверов Freenode, в других сетях они могут отличаться. Теперь ваш пользователь зарегистрирован на сервере IRC – то есть никто другой это имя не отберет. Сказать IRC-серверу, кто вы такой, можно с помощью команды:

```
/msg nickserv identify <password>
```

Нужно также зарегистрировать как пользователя нашего, пока нерожденного, бота. Переключитесь в другого пользователя и задайте ему пароль таким образом:

```
/nick UltraCoolLXFBot
/msg nickserv register thisismypassword
```

Конечно, можно взять другое имя пользователя и пароль, но запомните их, ведь они понадобятся коду нашего бота! Итак, мы завели две зарегистрированные учетные записи, себе и боту.



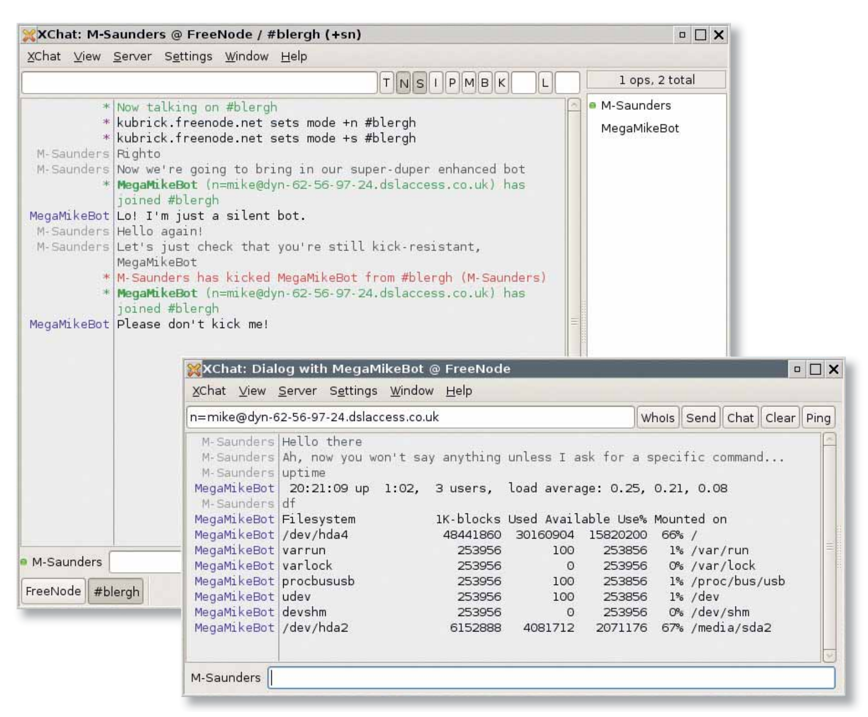
› Первая версия бота: по нашей просьбе он выдает информацию о времени работы системы. Но он способен на большее...

Покажи мне код

Мы готовы к запуску бота. Вот его код. Его не придется набирать вручную: он приведен на нашем **DVD** в разделе **Magazine/Perl** (файл **bot1.pl**). Но прежде чем запускать бота, не мешает просмотреть код и его описание...

```
use Net::IRC;
$server = 'irc.freenode.net';
$channel = '#blergh';
$botnick = 'MegaMikeBot';
$password = 'foobar';
$botadmin = 'M-Saunders';
$irc = new Net::IRC;
$conn = $irc->newconn(Nick => $botnick,
Server => $server, Port => 6667);
$conn->add_global_handler('376', \&on_connect);
$conn->add_global_handler('disconnect', \&on_disconnect);
$conn->add_global_handler('kick', \&on_kick);
$conn->add_global_handler('msg', \&on_msg);
$irc->start;
sub on_connect {
  $self = shift;
  $self->privmsg('nickserv', "identify $password");
  $self->join($channel);
  $self->privmsg($channel, "Lo! I'm just a silent bot.");
}
sub on_disconnect {
  $self = shift;
  $self->connect();
}
sub on_kick {
  $self = shift;
  $self->join($channel);
```

»



› Бот 2.0 теперь умеет выполнять заданные команды, и не станет говорить ни с кем, кроме вас!

```
$self->privmsg($channel, “Please don’t kick me!”);
}

sub on_msg {
  $self = shift;
  $self->privmsg($botadmin, `uptime`);
}
```

Давайте пройдемся по коду. Первая строка просто сообщает интерпретатору Perl, что мы хотим использовать ранее установленную библиотеку *Net::IRC*. На пяти следующих строках объявляются параметры конфигурации – перед запуском скрипта их потребуется изменить. Мы определяем адрес IRC-сервера, с которым соединится наш бот, и канал, в который он войдет; затем задаем имя пользователя (ник) и пароль, с помощью которых бот авторизуется на IRC-сервере. Наконец, задается имя пользователя, с которым бот будет говорить – ваша обычная учетная запись IRC.

Стоп... а что такое IRC?

IRC (Internet Relay Chat - ретранслируемый интернет-чат) - это система обмена сообщениями в режиме реального времени, предшественница всемирной паутины (World Wide Web), изобретенная в 1988 году. IRC - открытый текстовый протокол, и написать клиента для него может каждый. IRC-обсуждения проходят в нескольких сетях; одни ориентированы на свободное ПО (irc.freenode.net), а другие - на игры.

Каждая IRC-сеть состоит из нескольких серверов, связанных друг с другом - можно выбрать как ближайший к вам географически, так и любой другой случайным образом. В любом случае вы будете частью одной и той же сети. Например, чтобы начать общаться в сети Freenode, найдите IRC-клиент (например, X-Chat), запустите его и введите:

```
/server irc.freenode.net
```

(команды IRC начинаются со слэша). После этого зарегистрируйте имя пользователя и пароль с помощью следующих команд:

```
/nick <mynickname>
/msg nickserv register <mypassword>
```

Потом можно зайти в «канал» (отдельная комната чата) с помощью команды:

```
/join #linuxformat
```

(Название канала начинается с символа решетки.) Теперь можете писать сообщения в общий чат или приватные сообщения отдельным пользователям (например, в X-Chat для этого кликните правой кнопкой на имени пользователя и выберите в меню «Открыть диалог»).

После этого в строке $irc = new Net::IRC; создается объект IRC-бота, с творчески выбранным именем $irc. Затем мы формируем объект $conn, который создает соединение с IRC-сервером на основе предоставленной ранее информации. Итак, на данном этапе у бота есть все необходимое для подключения к IRC-серверу.

Но сперва нужно сделать еще одну вещь: связать с сообщениями, которые бот получает от IRC-сервера, некоторые действия. Наш бот ведь не глухонемой, и он должен знать, на какие сообщения давать ответ. Четыре вызова add_global_handler об этом позаботятся – они говорят боту: «Получив сообщение A, перейди к подпрограмме B».

Например, первая строка этого куска кода говорит: «Если получим сообщение 376, вызываем функцию on_connect». Сообщение 376 – специальный код, возвращаемый IRC-сервером на стадии установки соединения: сервер доставил информацию, что связь успешно установлена, и готов к приему команд. Если вы любитель посидеть в IRC, то знаете, что во время установки соединения на экране появляется множество сообщений о статусе сервера, сообщений данного сообщества и т.д. Код 376 означает конец приветствий.

Далее следуют обработчики еще для трех событий: разрыв соединения, удаление из чата и приватное сообщение. Обработчики говорят боту, какие функции вызвать в тех случаях, когда связь прекращается, бота гонят из чата или он получает приватное сообщение соответственно. IRC-боты могут реагировать и на всякие другие сообщения, но для нашего случая этого достаточно.

Команда $irc->start; приводит бота в действие. Бот обучен, какой сервер и канал использовать и какие подпрограммы вызывать при получении определенных сообщений; он соединяется с сервером и через несколько секунд получает сообщение 376, о котором мы говорили ранее. Оно активирует функцию on_connect.

В этой функции вы увидите команду shift, которая была описана в разделе с примером на Perl. Можете не вникать в суть кода, вкратце скажем, что он делает: аргументы передаются функции в массиве, и первый из них – вызывающий объект. Значение первого аргумента, вызывающий объект, присваивается переменной $self, тогда можно будет вызывать его функции. Есть и другие варианты, но наш обеспечивает модульность кода.

После соединения бота с сервером мы отправляем приватное сообщение (privmsg) nickserv – это обработчик имени пользователя на большинстве серверов IRC. Мы предоставляем пароль бота для аутентификации его на сервере. Затем мы заходим в заданный канал и отправляем небольшое сообщение, означающее, что мы будем вести себя тихо, не засоряя канал всяким хламом!

Теперь бот будет смирно сидеть в канале, пока его оттуда не пнут или не прервется соединение. В этих случаях он вызовет соответствующие функции и кротко попросит: «Пожалуйста, не удаляйте меня» (“Please don’t kick me!”) или повторно создаст соединение. Самая важная функция здесь – это on_msg. Она вызывается, когда бот получает приватное сообщение – то есть сообщение в отдельном диалоге вне основного канала. В ответ на любое сообщение бот отправляет вывод команды `uptime` пользователю $botadmin, т.е. вашему обычному пользователю.

Обратите внимание на обратные одиночные кавычки (`) вокруг команды `uptime`. Они велят Perl выполнить указанную команду [в оболочке, – *прим.ред.*] и вернуть результат в виде строки. Таким образом, строка с privmsg отправляет нашему пользователю приватное сообщение, содержащее вывод команды uptime. На большинстве клавиатур клавиша одиночной обратной кавычки находится под клавишей Escape.

Продолжаем работать

Пора запустить бота. Откройте файл **bot1.pl** и измените пять конфигурационных переменных сверху в соответствии с вашими настройками – а именно, впишите имя пользователя и пароль для бота, вами зарегистрированные, а свое имя пользователя присвойте переменной $botadmin. Запустите IRC-клиента, авторизуйтесь на сервере и зайдите в канал, который вы указали для бота. Затем запустите бота:

А вы что нараБОТали?

Написали своего бота с помощью этой статьи? Расскажите нам об этом: отправьте свой код по адресу letters@linuxformat.ru, и возможно, мы поместим его на LXFDVD. Если не удается заставить бота работать, загляните на наши форумы www.linuxforum.ru.

```
perl bot1.pl
```

Через пару секунд бот установит соединение, авторизуется на сервере и зайдет в указанный канал. Вы узнаете о его появлении по небольшому приветствию! Командой privmsg начните приватный диалог с ботом и скажите ему все, что угодно – бот пришлет результат выполнения команды uptime. Отлично! Теперь можно запустить бота на любом компьютере и отслеживать время работы системы (и уровни загрузки), просто общаясь с ботом в IRC-чате.

Хотя нарекания остаются. Кто бы ни заговорил с ботом в приватном канале, результат выполнения команды uptime отправится вам. Бот также умеет выполнять только команду uptime – может, вам этого и достаточно, но он способен на гораздо большее.

Если все работает нормально, можно добавить в последнюю функцию (on_msg) новые возможности, например, таким образом (полный код бота – в файле **bot2.pl** на DVD):

```
sub on_msg {
 $self = shift;
 $event = shift;
 if ($event->nick eq $botadmin) {
  foreach $arg ($event->args) {
   if ($arg =~ m/uptime/) {
    @output = `uptime`;
    foreach $line (@output) {
     $self->privmsg($botadmin, $line);
    }
   }
   if ($arg =~ m/df/) {
    @output = `df`;
    foreach $line (@output) {
     $self->privmsg($botadmin, $line);
    }
   }
  }
 }
}
```

На этот раз мы проверяем (if), кто отправляет боту приватное сообщение (третья строка), и продолжим только если это $botadmin, то есть вы! Посмотрим на текст, отправленный боту. Он содержится в массиве $event->args.

Пройдемся по тексту и выясним, есть ли там команда, подлежащая выполнению. Этим занимается строка if ($arg =~ m/uptime/): она означает «Если одна из строк текста, отправленного боту, содержит строку uptime, сделай следующее...» Фрагмент m/uptime/ – регулярное выражение, проверяющее, есть ли в тексте строка uptime.

Если она есть, запустим команду uptime и запишем результат ее выполнения в массив @output. Потом в цикле выведем каждую строку текста, содержащегося в массиве.

Uptime возвращает только одну строку текста – может, не стоит связываться с циклом? Но в следующем фрагменте кода проверяется, отправили ли мы команду df – а df возвращает несколько строк. Получив команду df, бот копит ее результат и отсылает его нам, строка за строкой.

Добавляйте новые команды, сколько душа просит. Для этого копируйте и вставляйте блок кода с if, а в нем меняйте текст, который там проверяется, и выполняемую команду. Например, можно добавить блок if ($arg =~ m/ps/), который получает список процессов с помощью команды @output = `ps`;. Можно вставить любую команду – бот все слопает!



Теперь у нас есть расширяемый IRC-бот, который отвечает на ваши команды и только вам. Он ни о чем не проболтается случайному посетителю (если только тот не зайдет под вашим именем и паролем). Вряд ли вы захотите разрешить боту выключать компьютер, но с его помощью можно получить о компьютере массу информации.

Это очень гибкое решение: если кажется, что с компьютером что-то случилось, можно запустить бота на web-сервере и допросить его; или – запустить бота на домашнем компьютере и прямо с работы запускать и останавливать с его помощью определенные процессы. Возможности безграничны – а всего-то нужен доступ к IRC! LXF

Скорая помощь

Масса возможностей модуля *Net::IRC* осталась за рамками этого обзора - просмотрите файлы **.pm** в архиве, чтобы узнать о них больше. Также рассмотрите другой пример реализации бота, скрипт в файле **irctest**.

Linux Format 97



При выборе IRC-клиента глаза разбежались? Пользователи Linux избалованы выбором!

Не страшно! В журнале под номером 97 мы испытывали восемь лучших: *KVIrc*, *Konversation*, *Kopete*, *XChat*, *Pidgin*, *Chatzilla*, *Opera* и *Irssi* прошлись перед нами строем ради определения победителя.

» Через месяц Изучите иностранный язык и подтяните географию с помощью магии Python.

Кластеры: повышаем

Отказоустойчивость – то свойство, которое помогает кластеру пережить сбой электропитания и другие бедствия. **Д-р Крис Браун** покажет, как удержаться на плаву.





Наш эксперт

Д-р Крис Браун
Пользователь Unix с более чем двадцатипятилетним стажем. Его компания Interactive Digital Learning предоставляет курсы, консультации и материалы для обучения по Linux. Он также специализируется на системах электронного обучения.

Журналы доступа к моей web-странице – не самое захватывающее чтиво. Если исключить случайные вторжения роботов поисковых систем, посетители заходят на сайт не чаще раза в неделю. Честно говоря, если сайт упадет, это заметят разве что через несколько месяцев. Даже грустно. Другая крайность – коммерческие web-сайты, приносящие владельцам ежедневный доход: несколько минут простоя такого сайта выливаются в ощутимые убытки для компании. На данном уроке я расскажу, как создать в Linux отказоустойчивый кластер. В случае краха основного сервера система автоматически переключается на запасной, а снаружи кластер выглядит как один очень надежный сервер. Нам поможет открытая программа *Heartbeat*, найти которую можно на сайте www.linuxha.org. Кластер мы построим из двух компьютеров (основной и запасной сервер). *Heartbeat* будет отвечать за обнаружение отказа основного сервера и управлять запуском нужного нам сервиса на запасном. Мы воспользуемся сервисом *Apache* (*httpd*), но эта технология подойдет и для любого другого сервиса: FTP, DNS или почтового. В общем, все будет работать примерно так:

1. В обычном режиме необходимые сервисы (типа *httpd*) предоставляются основным сервером.
2. Основной и запасной серверы постоянно обмениваются друг с другом сообщениями '*Heartbeat*' [англ. пульс], которые говорят запасному серверу, что с основным все в порядке.
3. Если запасной сервер не получает сообщения, он перенимает ресурсы основного сервера (например, запускает собственную копию сервера *httpd*).
4. Запасной сервер перенимает IP-адрес, по которому предоставлялся сервис на основном сервере.

Начинаем: компоненты системы

Если вы намерены двинуться дальше и попробовать все на практике, то вам понадобятся три компьютера – основной сервер, запасной сервер и клиент, с помощью которого мы сможем проверить, как все работает.

На основном и запасном серверах должен быть установлен Linux, на клиенте – необязательно (хотя под Linux в вашем распоряжении окажутся лучшие средства диагностики). Компьютеры должны быть подключены к сети Ethernet. Для обмена сигналами *Heartbeat* на основном и запасном серверах должна быть вторая сетевая карта. Эти карты нужно соединить друг с другом либо перекрестным (crossover) кабелем, либо через мини-хаб. Если на обоих компьютерах есть последовательный порт, то вместо второго Ethernet-подключения можно соединить их последовательные порты перекрестным кабелем, по которому будет производиться обмен сигналами. Общая схема показана на **Рис. 1**.

Установка heartbeat-соединения

Если обмен сигналами *Heartbeat* будет осуществляться через Ethernet, соединить сетевые карты друг с другом можно двумя способами: либо через перекрестный кабель RJ45, либо парой обычных кабелей через мини-хаб. Первый вариант лучше, потому что здесь нечему выходить из строя. Каждой из карт нужно назначить статический IP-адрес, причем выбранный из частного диапазона, ни один адрес из которого не используется во внутренней сети. Например, первой карте можно назначить адрес 10.0.0.1, второй – 10.0.0.2. Убедитесь, что компьютеры могут достучаться друг до друга [ping].

Также можно использовать и соединение через последовательный порт. Тогда нам потребуется последовательный перекрестный кабель. На обоих концах этого кабеля, который еще называют нуль-модемным, имеется 9-контактный разъем типа «мама». Кабелем нужно соединить последовательные порты компьютеров. (Последовательные порты и кабели сейчас используются редко, и если на ваших компьютерах их нет, проще пойти и купить вторую сетевую карту для соединения по Ethernet.)

Чтобы проверить соединение через последовательные порты, выполните на основном сервере команду:

```
cat /dev/ttyS0
```

а на запасном – команду:

```
cat /etc/fstab > /dev/ttyS0
```

На основном сервере должно отображаться содержимое файла. Повторите эксперимент в обратном направлении.

Heartbeat будет прекрасно работать с соединением любого типа. Преимущество Ethernet-соединения – трафик *Heartbeat* в этом случае можно отслеживать с помощью одной из утилит для отслеживания пакетов (например, wireshark). Можно даже использовать одновременно оба соединения, что позволит обойтись без ненужных откатов, если только одно из них выйдет из строя.

Месяц назад Ядро: собери себе свое. Мы научились накладывать патчи.

отказоустойчивость

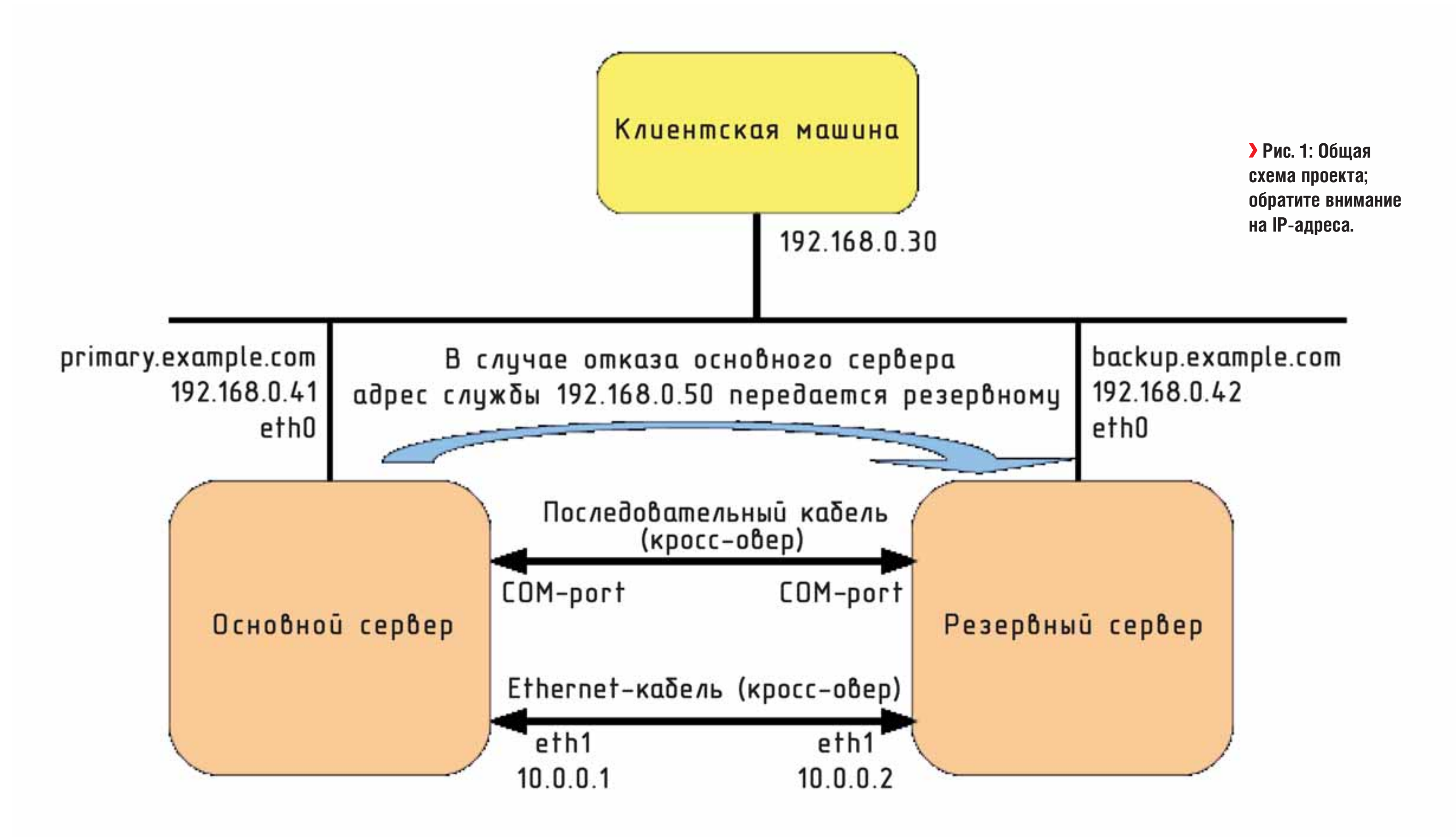


Рис. 1: Общая схема проекта; обратите внимание на IP-адреса.

Настройка сетевых интерфейсов

Оба сетевых интерфейса, которые участвуют в *Heartbeat*-соединении, должны быть настроены на статические IP-адреса, так как DHCP-сервера в этой «сети» точно нет. Статические адреса должны быть и на основных интерфейсах, которые используются для подключения к основному и запасному серверам. Обратите внимание, что это НЕ те адреса, на которых предоставляется сервис. *Heartbeat* назначит сетевой карте исправного компьютера второй IP-адрес, и именно он будет использоваться клиентами для доступа к ресурсу.

Три файла конфигурации

Сразу после установки *Heartbeat* нужно создать в каталоге **/etc/ha.d** три (да, три!) файла конфигурации, а именно:

» **ha.cf** Файл содержит настройки самого *Heartbeat*; в нем определяются используемые программой пути и устанавливаются некоторые временные параметры.

» **haresources** Этот файл определяет ресурсы (сервисы), которыми будет управлять *Heartbeat*, и основной сервер для каждого из этих ресурсов.

» **authkey** В этом файле задается пароль и метод шифрования, используемый для аутентификации *Heartbeat*-сообщений.

Начнем с **ha.cf**. Пример этого файла можно найти в документации на *Heartbeat* (**/usr/share/doc/heartbeat/ha.cf**). Это один из файлов типа «визгу много, проку мало», которые целиком состоят из документации и закомментированных примеров. Можно скопировать его в **/etc/ha.d** и использовать как отправную точку, или создать свой файл с нуля.

Вот минимальный вариант этого файла. Номера строк даны только для удобства ссылок, в самом файле их нет:

```
1. udpport 694
2. bcast eth1
3. serial /dev/ttyS0
4. baud 9600
5. keepalive 2
6. deadtime 20
7. initdead 60
8. node primary.example.com
9. node backup.example.com
```

Строки 1 и 2 задают номер порта и сетевой интерфейс, используемые для *Heartbeat*-соединения. Для соединения через последова- »

Ошибка!

В этой статье мы не говорим о том, как предотвратить неисправности других типов, например, те, что возникают в инфраструктуре самого интернета. В частности, мы не говорим о создании дополнительных информационных каналов для связи с сайтом. Впрочем, это не так уж и важно, потому что врожденная гибкость архитектуры интернета позволяет поглощать частные сбои. Часто даже говорят, что сеть ARPANET (предшественник интернета) разрабатывалась с таким расчетом, чтобы выдержать взрыв атомной бомбы. ARPANET и правда должна была справляться с утратой части сети, но главная причина была в том, что коммутация узлов сети и связей внутри нее была не особенно надежной, даже и без атомной бомбы.

» тельные порты в строках 3 и 4 задается порт и скорость соединения. Можно (и рекомендуется) использовать оба типа, но у меня было только соединение через последовательные порты, и я закомментировал строки 1 и 2. В строке 5 задается тайм-аут между сообщениями, а строка 6 определяет время, по истечении которого, не дождавшись очередного сообщения от основного сервера, запасной сочтет его покойником. В строке 7 указывается время, необходимое для активации сетевого интерфейса после включения компьютера, после чего *Heartbeat* начинает функционировать. В строках 8 и 9 задаются имена хостов для основного и запасного серверов (те, что возвращает команда uname -n).

Для начала попробуем управлять полностью фиктивным сервисом **tester**. Мы напишем скрипт, **/etc/ha.d/resources.d/tester**, вызываемый *Heartbeat*. Он будет всего лишь вызывать команду logger, которая сообщит нам (через *syslogd*), что скрипт был запущен. Вот как он выглядит:

```
#!/bin/bash
logger $0 called with argument $1
```

Скрипт нужно сделать исполняемым, например, так:

```
chmod 755 tester
```

В следующем файле, *haresources*, содержится список ресурсов, которыми управляет *Heartbeat*. Когда компьютер выходит из строя, эти ресурсы перемещаются с одного узла сети на другой. (попробуйте произносить это как “ha-resources”, а не “hare-sauces” [англ. hare – заяц, source – источник, sauce – соус, – *прим. ред.*]. По правде говоря, я сроду не ел зайчатины, но первым делом подумал о кролике и решил, что подойдет клюквенный соус.) Опять же, пример этого файла можно найти в документации (**/usr/share/doc/Heartbeat**) – можете просто скопировать его в **/etc/ha.d** или создать свой собственный файл с нуля. Для нашего первого теста в этом файле должна быть всего одна строка:

```
primary.example.com tester
```

Также нужно создать файл **/etc/ha.d/authkeys**, содержащий ключи, используемые для аутентификации *Heartbeat*-сообщений, например:

```
auth1
1 sha1 foobar
```

Нужно позаботиться, чтобы читать этот файл мог только администратор:

```
chmod 600 authkeys
```

Если права доступа к этому файлу есть у кого-то еще, *Heartbeat* не запустится.

Скопируйте эти четыре файла с основного сервера на запасной в соответствующие каталоги. Файл **ha.cf**, возможно, придется подправить в зависимости от аппаратного обеспечения компьютера, а остальные файлы должны быть одинаковыми на обоих компьютерах.

Теперь можно запустить *Heartbeat*. Для отслеживания сообщений *Heartbeat* на обоих компьютерах запустите tail -f для файла журнала (**/var/log/messages**). Запустите программу с помощью команды:

```
/etc/init.d/heartbeat start
```

Украденная личность

Перенос IP-адреса на другой компьютер – компьютерный эквивалент «воровства личности»: вы притворяетесь кем-то другим, а потом убеждаете весь мир, что вы настоящий. В процессе переноса системы на запасной сервер обязано произойти следующее. Во-первых, основной сервер должен отдать IP-адрес. Если он загорелся или нашел другую кончину, это произойдет по умолчанию, но если *heartbeat* останавливается на другом (исправном) сервере, адрес, очевидно, нужно забрать назад. Другая проблема – обновление записей в ARP-кэше. Доставка сетевого пакета на конкретный компьютер в конечном счете основана на его MAC-адресе – уникальном 48-битном номере, зашитом в каждую сетевую карту. Чтобы определить MAC-адрес компьютера по его IP-адресу, используется протокол ARP (Address Resolution Protocol – протокол разрешения адресов). Записи о соответствии IP-адресов и MAC-адресов обычно хранятся в кэше ARP 10–15 минут. Чтобы принудительно обновить ARP-кэш на других компьютерах, heartbeat рассылает в сеть так называемые «незаконные» [gratuitous] ARP-пакеты с компьютера, перенявшего IP-адрес.

Загружаем heartbeat

Heartbeat нужно установить и на основном, и на запасном сервере. У меня на обоих компьютерах была установлена CentOS 5 (фактически, RHEL5). Я просто установил *heartbeat* вместе с зависимостями (которые включали *heartbeat-pils* и *heartbeat-stonith*) из репозиториев CentOS с помощью *yum*. Похожие пакеты есть и в репозиториях других дистрибутивов, таких как Ubuntu; в виде пакета RPMs для SUSE/Novell Linux их можно найти на сайте **www.linux-ha.org** или загрузить архив с исходными кодами с этого же сайта и скомпилировать программу из них.

На обоих компьютерах действия *Heartbeat* будут писаться в журнал. На основном сервере среди прочего должно появиться сообщение типа:

```
logger: /etc/ha.d/resource.d/tester called with argument start
```

Оно означает, что основной сервер, как и ожидалось, запустил сервис **tester**.

Heartbeat запущен, и мы можем проверить поведение системы в случае отказа. Я просто выдернул шнур электропитания из основного сервера и наблюдал за сообщениями в файле журнала запасного. По истечении заданного времени запасной сервер сообщил об отказе основного и запустил сервис **tester**. Вот соответствующие строки журнала:

```
heartbeat: [9615]: WARN: node primary.example.com: is dead
heartbeat: [9615]: WARN: No STONITH device configured.
heartbeat: [9615]: WARN: Shared disks are not protected.
heartbeat: [9615]: info: Resources being acquired from primary.example.com.
heartbeat: [9615]: info: Link primary.example.com:/dev/ttyS0 dead.
heartbeat: [10305]: info: No local resources [/usr/share/heartbeat/ResourceManager listkeys backup.example.com] to acquire.
harc[10304]: [10323]: info: Running /etc/ha.d/rc.d/status status
mach_down[10329]: [10350]: info: Taking over resource group tester
ResourceManager[10351]: [10362]: info: Acquiring resource group: primary.example.com tester
logger: /etc/ha.d/resource.d/tester called with argument status
ResourceManager[10351]: [10379]: info: Running /etc/ha.d/resource.d/tester start
logger: /etc/ha.d/resource.d/tester called with argument start
mach_down[10329]: [10384]: info: mach_down takeover complete for node primary.example.com.
```

Первая проверка завершилась успешно, и пора поручить *Heartbeat* управление полноценным сервисом. Для этого проекта я выбрал *httpd*. Таким образом, сейчас нам нужно убедиться, что *Apache* установлен на обоих компьютерах, и положить в корневой каталог web-сервера какие-нибудь легко распознаваемые файлы. Я просто создал два файла **/var/www/html/index.html**, по одному для основного и запасного серверов; в файле для основного сервера была строка “This is the website from the primary server”, для запасного – “This is the website from the backup server” («Это сайт с основного/запасного сервера»). На обоих компьютерах запустите *httpd*:

```
/etc/init.d/httpd start
```

и проверьте, что, открыв в браузере адрес **http://localhost**, вы видите эти файлы. Теперь остановите *Apache* и убедитесь, что он не запустится автоматически при загрузке системы. Воспользуйтесь командами:

```
/etc/init.d/httpd stop
chkconfig httpd --del
```

Это важно, потому что *Apache* должен запуститься не во время загрузки, а только если *Heartbeat* запустил его.

Теперь нужно изменить файл *haresources* так:

```
primary.example.com httpd
primary.example.com 192.168.0.50
```

В первой строке указывается сервис, которым будет управлять *Heartbeat*, а во второй – IP-адрес, используемый клиентом для доступа к сервису. При запуске *Heartbeat* этот адрес назначается на основной сервер как второй IP-адрес. Если основной сервер откажет, копия

Heartbeat, запущенная на запасном сервере, перенесет этот IP-адрес на свой сетевой интерфейс.

Предполагается, что *Heartbeat* будет управлять сервисами путем запуска скрипта, указанного в **haresources**, с аргументами start, stop или status. Конечно, это те самые аргументы скриптов каталога **/etc/init.d**, что используются для запуска сервисов во время загрузки системы, и, по правде говоря, *Heartbeat* будет автоматически искать скрипт **httpd** в каталоге **/etc/init.d**.

После внесения этих изменений в конфигурацию основного и запасного серверов можно снова запустить *Heartbeat* на каждом из них:

```
/etc/init.d/heartbeat start
```

Когда *Heartbeat* войдет в нормальный режим работы, нужно кое-что проверить. На основном сервере запустите:

```
ip addr show
```

и проверьте, что *Heartbeat* назначил второй IP-адрес (в нашем примере 192.168.0.50) на наш сетевой интерфейс. Также запустите команду

```
ps -ef | grep httpd
```

которая проверит, что сервис *httpd* запущен. На запасной сервер второй IP-адрес не назначается, а сервис *httpd* на нем не выполняется.

Теперь пора загрузить третий компьютер, чтобы использовать в качестве клиента. Убедитесь, что он может разрешить имя компьютера, на котором запущен сервис, по его IP-адресу. Для этого я просто добавил строку

```
192.168.0.50 www.example.com
```

в файл **/etc/hosts**. Проверьте, пингуется ли адрес www.example.com с компьютера клиента. Откройте в браузере на клиентском компьютере страницу http://www.example.com. Вы должны увидеть файл **index.html** основного сервера. На клиенте также можно проверить кэш **arp**, запустив команду

```
arp -a
```

и обратить внимание на MAC-адрес, связанный с IP-адресом 192.168.0.50. В моем случае результат работы команды был таким:

```
www.example.com (192.168.0.50) at 00:0C:F1:96:A3:F7 [ether] on eth0
```

Ладно, и как это работает?

Мы подошли к драматической развязке: демонстрации переноса сервиса на запасной сервер в случае отказа основного. Можно было просто отключить питание основного сервера, как и в предыдущем эксперименте, но на сей раз я избрал более кроткий способ – просто остановил *Heartbeat* на основном сервере командой:

```
/etc/init.d/heartbeat stop
```

Остановившись, *Heartbeat* остановит *httpd* на основном сервере и проинформирует запасной сервер об отключении основного (через *Heartbeat*-соединение). Запасной сервер сразу возьмет управление на себя, и в файле его журнала (**/var/log/messages**) появятся новые сообщения. Два из них наиболее важны (время и другую информацию в начале строки я удалил):

```
Running /etc/init.d/httpd start
Registering new address record for 192.168.0.50 on eth0
```

В этом случае все заняло пару секунд. Если бы я просто выдернул из сети шнур питания основного сервера, то на восстановление системы потребовалось бы больше времени: в этом случае перед тем, как сделать вывод об отказе основного сервера и взять управление на себя, запасной сервер ожидал бы ответа 20 секунд (параметр deadtime в файле **ha.cf**).

На клиентском компьютере следует провести пару важных тестов. Во-первых, попробуйте обновить страницу www.example.com в браузере. Должна появиться страница запасного сервера. Во-вторых, еще раз просмотрите содержимое ARP-кэша. На моем компьютере команда arp –a вернула следующий результат:

```
www.example.com (192.168.0.50) at 00:10:60:60:3E:8E [ether] on eth0
```

Это означает, что пакеты, предназначенные для адреса 192.168.0.50, теперь будут направляться на запасной сервер.

А что же дальше? Ну, наверное, со временем кто-то доберется до основного сервера, починит его и перезапустит. Когда *Heartbeat* перезапускается, он сообщает об этом запасному серверу через *Heartbeat*-соединение. В журнале запасного сервера появятся такие строки:

```
Heartbeat restart on node primary.example.com
Releasing resource group: primary.example.com httpd
Running /etc/init.d/httpd stop
Withdrawing address record for 192.168.0.50 on eth0
```

Они означают, что запасной сервер освободил ресурсы и разрешил основному забрать их.

Если у вас хватило духу дочитать до конца, поздравляю! Вы создали отказоустойчивый кластер в Linux и можете начать делать на этом деньги в реальном мире! Однако прежде чем пойти на собеседование в Google, NASA или даже Krazy Ken's 24x7 Liquorice Emporium, быть может, вы захотите узнать о некоторых вопросах подробнее.

Один из них – это концепция «ограждения» (fencing). Когда запасной сервер перенимает ресурсы основного, нужно как-то гарантировать, чтобы основной сервер не попытался снова предоставить эти сервисы. *Heartbeat* может использовать для этого механизм STONITH (“shoot the other node in the head” – «контрольный выстрел в голову»), если вы купите устройство, программно отключающее питание компьютера. Второй вопрос – как синхронизировать содержимое двух web-серверов. В нашем примере мы умышленно сделали его разным, но на практике может потребоваться какое-то средство синхронизации их содержимого (*rsync* или *unison*). LXF

Дополнительная информация

Побольше узнать о *heartbeat* можно в документации **/usr/share/doc/heartbeat** (каталог зависит от конкретного дистрибутива), в частности, в файле **GettingStarted**. На сайте Linux HA также есть руководства и документация. Рекомендую вам прекрасную книжку “Linux Enterprise Cluster: Build a Highly Available Cluster with Commodity Hardware and Free Software” (No Starch Press, ISBN 1-59327-036-4).



› Мы думали, что саморекламы nostarch.com как «лучшего развлечения для хакеров»™ — это гипербола… Но нет! Вы только взгляните на заголовки!



› Сайт Linux HA (www.linux-ha.org) доступен даже на японском. Огромное количество ссылок в разделе *Press* наглядно свидетельствует о популярности, завоеванной проектом с момента его основания.

Через месяц Распределите нагрузку на виртуальный сервер по нескольким реальным.

Очепятки не пройдут

Пытаясь набрать это предложение, наш редактор допустил три описки. И что бы он делал без модуля проверки орфографии? **Петр Семилетов** расскажет, как прикрутить такой к вашей программе.



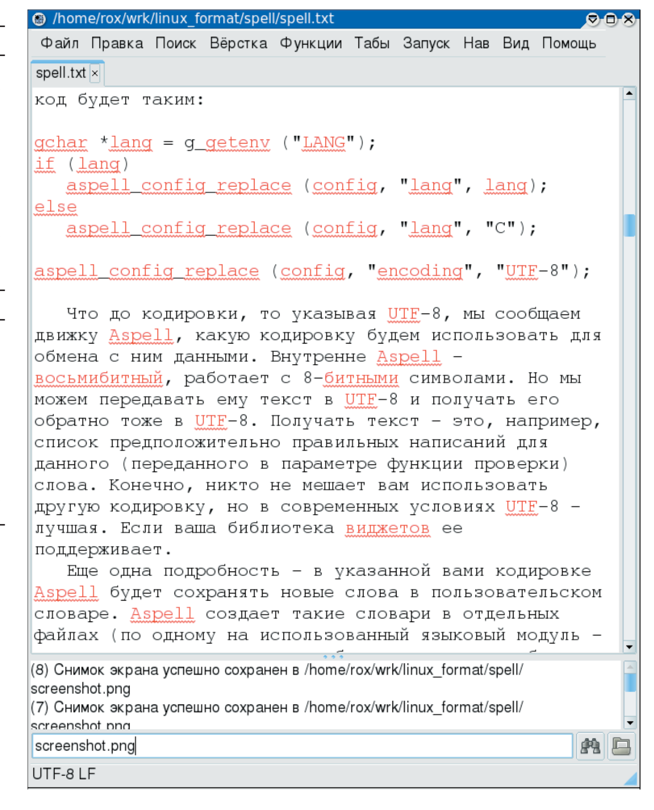
Наш эксперт

Петр Семилетов
Разработчик свободного текстового редактора TEA, известного своими уникальными функциями.
tea-editor.sf.net

Потребность в движках проверки орфографии растет с развитием общения через Интернет. Если раньше ошибки правописания подсвечивались разве что в текстовых редакторах и процессорах, то нынче это делают и браузеры, и программы для обмена мгновенными сообщениями. Сложно сказать, улучшает ли компьютерная проверка грамотность человека. С одной стороны, пользователь сразу видит свои ошибки и в следующий раз может написать слово правильно. Однако, полагаясь в грамотности на машину, не потакает ли человек лени своего разума?

На этом уроке мы не будем задаваться столь философскими вопросами, а просто рассмотрим, как задействовать движок проверки правописания *Aspell* в своих программах. Мы будем использовать API для языка C, а для большей наглядности практического применения ряд примеров будет дан с привязкой к *GTK+* (мы публиковали серию статей о нем в период с LXF86 по LXF95, так что вы найдете все эти уроки на **LXFDVD**), однако, изложенные общие принципы работы применимы к любой библиотеке. В конце материала будет уделено немного места еще одному движку – *Enchant*, для сравнения. Также обратите внимание на *Hunspell* – он уже используется в *OpenOffice.org*, а со временем должен заменить *MySpell* и в продуктах *Mozilla*.





ЧАСТЬ 1: GNU Aspell

Aspell (http://aspell.net) – один из самых популярных движков проверки орфографии. Он состоит из консольной программы (которой, помимо прочего, можно передавать данные через канал), а также библиотеки с «сишным» API, хотя сам *Aspell* написан на C++. Работы с каналами мы касаться не будем.

Описанию API в документации *Aspell* уделено мало внимания – несколько примеров да примечания. Чтобы использовать *Aspell* в своей программе, надо изучить его исходные тексты – причем не только заголовочные файлы. Но сначала не будет излишним прочесть эту статью.

Начнем с проверки, установлена ли *LibAspell* в системе пользователя, и включения ее в параметры компилятора. По какой-то причине библиотека не оснащена модулем для **pkg-config**, поэтому искать pc-файл бесполезно. Вместо этого, в случае Autotools, следует добавить в файл **configure.in** (который будет обработан **autogen.sh** для создания скрипта *configure*) макросы для проверки наличия заголовочного файла **aspell.h**:

```
AC_CHECK_HEADER(aspell.h,
    LIBS="$LIBS -laspell"
    AC_DEFINE(HAVE_LIBASPELL, 1, [use aspell]),
        [AC_MSG_ERROR([cannot find header for libaspell])]
    )
```

Напомню формат макроса AC_CHECK_HEADERS:

```
AC_CHECK_HEADERS (имя заголовочного файла, [действие в положительном случае], [действие в отрицательном])
```

Итак, мы проверяем, есть ли в системе файл **aspell.h**. Если есть, то в файле **config.h** определяется флаг HAVE_LIBASPELL, чтобы потом в коде программы мы могли написать что-то вроде:

```
#ifdef HAVE_LIBASPELL
#include "aspell.h"
#endif
```

В приведенном выше примере, для вывода сообщения об ошибке был использован макрос AC_MSG_ERROR. Помимо прочего, он вызывает прекращение работы сценария configure. То есть, если **aspell.h** не

была найден, то и настройка исходных текстов будет провалена. Если такое поведение нежелательно, а наличие **aspell.h** – необязательное условие для сборки вашей программы, то проверка может выглядеть немного иначе:

```
AC_CHECK_HEADER(aspell.h,
            LIBS="$LIBS -laspell"
            AC_DEFINE(HAVE_LIBASPELL, 1, [use aspell]),
                        echo "aspell.h not found"
            )
```

Здесь, вместо использования сурового макроса мы просто выводим сообщение, что **aspell.h** не найден.

Приступим к работе с библиотекой. Первым делом создадим экземпляр класса AspellConfig. Класс этот служит для управления всякими настройками. Именно всякими, поскольку он не входит в инкапсуляцию других классов *Aspell*, но может быть применен для изменения настроек других классов. Создадим экземпляр:

```
AspellConfig *config = new_aspell_config ();
```

После окончания работы его надо будет удалить при помощи функции delete_aspell_config(). Значения полей в классе можно менять при помощи функции aspell_config_replace(). Формат вызова таков:

```
aspell_config_replace (config, имя переменной-поля, новое значение);
```

Давайте для примера настроим класс под русскую локаль и кодировку UTF-8:

```
aspell_config_replace (config, "lang", "ru");
aspell_config_replace (config, "encoding", "UTF-8");
```

Замечу, что у пользователя может быть другая локаль, тогда ваше 'ru' будет бесполезным. Поэтому лучше определить локаль программно. Для этого нужно прочитать значение переменной окружения 'LANG;. В *GTK+* это делается вот так:

```
gchar *lang = g_getenv ("LANG");
if (lang)
  aspell_config_replace (config, "lang", lang);
else
  aspell_config_replace (config, "lang", "C");

aspell_config_replace (config, "encoding", "UTF-8");
```

Если, вдруг, переменная 'LANG' не установлена, то мы используем значение 'C' (стандартная англоязычная локаль). Что до кодировки, то указывая UTF-8, мы сообщаем движку *Aspell*, какую кодировку будем использовать для обмена данными с ним. Изнутри *Aspell* восьмибитный, но мы можем передавать ему текст в UTF-8 и получать его обратно тоже в UTF-8. Полученным текстом может быть, например, список предположительно правильных написаний для данного (переданного в параметре функции проверки) слова. Конечно, никто не мешает вам использовать другую кодировку, но в современных условиях UTF-8 – лучшая, если ваша библиотека виджетов ее поддерживает.

Еще одна подробность – в указанной вами кодировке *Aspell* будет сохранять новые слова в пользовательском словаре. *Aspell* создает такие словари в отдельных файлах (по одному на каждый использованный языковый модуль – то есть на тот модуль, куда было виртуально добавлено слово). Файлы эти лежат в домашнем каталоге пользователя. Например, имя файла с пользовательским словарем для русского языка – **~/.aspell.ru.pws**.

Обратите внимание, что после каждой смены в движке текущего языка надо снова задавать кодировку, иначе движок будет работать в кодировке по умолчанию. Если явно не указать кодировку для русского, ею будет KOI8-R.

Чтобы обеспечить программу возможностью использовать любой доступный модуль проверки орфографии, необходимо получить список установленных модулей. В приведенном ниже примере мы получаем такой список и выводим имена модулей в консоль:

```
const AspellDictInfo *entry;
AspellConfig *config = new_aspell_config ();
AspellDictInfoList *dlist = get_aspell_dict_info_list (config);

AspellDictInfoEnumeration *dels = aspell_dict_info_list_elements (dlist);

while ((entry = aspell_dict_info_enumeration_next (dels)) != 0)
    if (entry)
      printf ("%s\n", entry->name);

delete_aspell_dict_info_enumeration (dels);
delete_aspell_config (config);
```

Итогом работы этого кода будет нечто вроде:

```
ru
en
en_CA
```

и так далее.

Полученные названия можно использовать в aspell_config_replace(), чтобы установить локаль движка *Aspell*:

```
aspell_config_replace (config, "lang", локаль);
```

Теперь у нас есть все знания для того, чтобы настроить язык и кодировку. Но этого мало. Надо создать на основе этих данных экземпляр класса, отвечающего за проверку орфографии. Для этого API предоставляет нам две функции, которые мы рассмотрим подробно:

```
struct AspellCanHaveError* new_aspell_speller (struct AspellConfig
*config);
struct AspellSpeller* to_aspell_speller (struct AspellCanHaveError *obj);
```

Внутри первой функции происходят любопытные вещи. *Aspell*, как уже упоминалось выше, написан на C++, и «сишной» структуре AspellCanHaveError соответствует внутренний класс CanHaveError, а AspellSpeller – класс Speller. Итак, функция new_aspell_speller() создает экземпляр класса Speller. В случае ошибки new_aspell_speller() возвращает экземпляр CanHaveError. В противном же случае возвращается Speller, которого нужно «вытащить» из CanHaveError функцией to_aspell_speller(). Я понимаю, что API можно было сделать более логичным.

Вот пример. Сначала вызываем new_aspell_speller:

```
AspellCanHaveError *possible_err = new_aspell_speller(spell_config);
```

На этом этапе мы не знаем, что вернулось в possible_err на самом деле – экземпляр AspellCanHaveError или AspellSpeller. Поэтому объявляем spell_checker и проверяем possible_err – то есть вернулась ли ошибка. Если да, то печатаем сообщение об ошибке, а если нет, то функцией to_aspell_speller приводим possible_err к типу AspellSpeller.

```
AspellSpeller *spell_checker;

if (aspell_error_number (possible_err) != 0)
   printf ("%s\n", aspell_error_message (possible_err));
else
   spell_checker = to_aspell_speller (possible_err);
```

Пара слов об освобождении памяти. Память для AspellCanHaveError освобождается с помощью функции delete_aspell_can_have_error(), а для AspellSpeller – delete_aspell_speller(). В приведенном выше коде, будь он в рабочей программе, освобождать память следовало бы так. Если вернулся экземпляр AspellSpeller, то память для AspellCanHaveError уже не нужно освобождать, потому что на самом-то деле в possible_err хранится не экземпляр AspellCanHaveError, а экземпляр AspellSpeller, и вызовы обеих функций уничтожения объекта приведут к ошибке сегментации во второй из них.

Лучший вариант работы с памятью таков. Если вернулся AspellCanHaveError – выполняем для него delete_aspell_can_have_error() и завершаем код проверки орфографии. Если вернулся AspellSpeller, то проверяем орфографию и очищаем память с помощью delete_aspell_speller(), однако delete_aspell_can_have_error() для possible_err уже НЕ вызываем.

Тонкая красная линия

Получив на руки работоспособный экземпляр AspellSpeller, мы, наконец, можем проверить написание слова. Для этого служит функция aspell_speller_check():

```
int aspell_speller_check (struct AspellSpeller *speller, const char *word,
int word_size);
```

Она возвращает нулевое значение, если слова нет в словаре, **1** – если есть, и **-1** в случае возникновения ошибки (в движке). То есть в рабочем коде, если нам нужно просто проверить, есть слово в словаре или нет, и нас не волнует возможная внутренняя ошибка, то можно писать так:

```
if (! aspell_speller_check (параметры)) {
  делаем что-то - например, подчеркиваем ошибочное слово
}
```

Рассмотрим параметры функции поподробнее. Со speller'ом все понятно. Слово word, передаваемое для проверки, должно быть в той кодировке, которую вы задали для словарного модуля. word_size – размер (в байтах) этого слова, может быть **-1**, если мы имеем дело со строкой, завершающейся нулем ('\0'). Обычно так оно и есть.

Попробуем *Aspell* в деле. Как проверить орфографию в стандартном виджете текстового редактора *GTK+* (да и в GtkSourceView, см. LXF97)?

Как вы, наверное, знаете, изменение атрибутов участков текста в GtkTextView осуществляется с помощью объектов-тэгов. Тэг несет в себе параметры шрифта, цвет и так далее. Тэги хранятся в таблице, которая подключается к GtkTextBuffer (именно к буферу, а не к GtkTextView). Сначала надо создать таблицу, а затем поместить в нее тэг.

Давайте создадим пустую таблицу:

```
GtkTextTagTable *tags_table = gtk_text_tag_table_new ();
```

Затем создадим тэг, которым будем подсвечивать ошибочные слова:

```
GtkTextTag *tag_spell_err = gtk_text_tag_new ("spell_err");
```

Обратите внимание – мы даем тэгу имя spell_err, чтобы потом иметь возможность обратиться к нему. Назначим тэгу следующие свойства: цвет букв (переднего плана, foreground) и подчеркивание (underline):

```
g_object_set (G_OBJECT (tag_spell_err), "foreground", "red", NULL);
g_object_set (G_OBJECT (tag_spell_err), "underline", PANGO_
UNDERLINE_NORMAL, NULL);
```

В данном примере мы использовали стиль PANGO_UNDERLINE_NORMAL – обычное подчеркивание прямой линией. Но в библиотеке Pango, начиная с версии 1.4, появился стиль PANGO_UNDERLINE_ERROR, созданный специально для подчеркивания ошибок (волнистая линия мелким зигзагом). Как написать код, проверяющий версию Pango и устанавливающий стиль подчеркивания в зависимости от нее? А вот так:

```
#if defined(PANGO_VERSION_CHECK) && PANGO_VERSION_
CHECK(1,4,0)
g_object_set (G_OBJECT (tag_spell_err), "underline", PANGO_
UNDERLINE_ERROR, NULL);
#else
g_object_set (G_OBJECT (tag_spell_err), "underline", PANGO_
UNDERLINE_SINGLE, NULL);
#endif
```

Теперь тэг можно поместить в таблицу:

```
gtk_text_tag_table_add (tags_table, tag_spell_err);
```

Создадим отдельный текстовый буфер с указанной таблицей тэгов:

```
GtkTextBuffer *text_buffer = gtk_text_buffer_new (tags_table);
```

У GtkTextView уже есть буфер по умолчанию. Заменим его на новый буфер с нашей таблицей тэгов:

```
gtk_text_view_set_buffer (text_view, text_buffer);
```

Альтернативный вариант вариант – не создавать свой буфер и таблицу, а получить указатель на уже существующий буфер с помощью функции gtk_text_buffer_get_tag_table(), и добавить тэг в полученную таблицу. Решение зависит от вас и от архитектуры вашей программы.

Приведем код, отвечающий непосредственно за проверку содержимого текстового буфера на ошибки:

```
GtkTextIter iter;
GtkTextIter a;
GtkTextIter b;

gchar *p = NULL;

gchar *text;
gtk_text_buffer_get_iter_at_offset (text_buffer, &iter, 0);

if (gtk_text_iter_starts_word (&iter))
   a = iter;

do
  {
   if (gtk_text_iter_starts_word (&iter))
     {
      b = iter;
      if (gtk_text_iter_backward_char (&b))
         a = iter;

      if (gtk_text_iter_forward_word_end (&iter))
         if (gtk_text_iter_ends_word (&iter))
           {
            text = gtk_text_iter_get_slice (&a, &iter);

            if (text)
              {
               if (g_utf8_strlen (text, -1) > 1)
                 {
                  if (! aspell_speller_check (text_buffer, text, -1))
                    {
                     gtk_text_buffer_apply_tag (text_buffer, gtk_text_tag_
table_lookup(gtk_text_buffer_get_tag_table (text_buffer), "spell_err"), &a,
&iter);
                     g_free (text);
                     continue;
                    }
                 }
               g_free (text);
              }
           }
     }
  }
while (gtk_text_iter_forward_char (&iter));
}
```

Неплохо, правда? Вначале мы получаем итератор *iter*, указывающий на начало буфера. Также у нас есть вспомогательные итераторы a и b, которых мы используем для более точного последовательного перебора букв в буфере. Перебирая в цикле один символ за другим, мы получаем слова. Текст каждого слова помещаем в переменную text и передаем эту переменную в функцию aspell_speller_check(). Если слова нет в словаре, то применяем цветовой тэг к участку в буфере, отмеченному итераторами, которые ограничивают текущее слово. Тэг применяется с помощью функции gtk_text_buffer_apply_tag(). Этой функции передаются следующие параметры: text_buffer – текстовый буфер, указатель на тэг. Его мы извлекаем по имени из таблицы тэгов, которая назначена данному буферу. Делается это так:

```
gtk_text_tag_table_lookup(gtk_text_buffer_get_tag_table (text_buffer),
"spell_err")
```

Оставшиеся два параметра – итераторы, задающие начало и конец отмечаемого тэгом участка в буфере. Вот вам задача для самостоятельного решения: перед вызовом кода подчеркивания ошибок надо убрать возможное прежнее подчеркивание, ведь есть вероятность, что пользователь уже проверял орфографию, а затем исправил некоторые слова. Поэтому, прежде чем заново подчеркивать ошибки, найдите в буфере все тэги с именем "spell_err" и удалите их.

Помощь зала

Помимо того, что *Aspell* может сообщить вам о наличии или отсутствии слова в словаре, он также дает список возможных вариантов верного написания предложенного слова. В приведенном ниже коде предпола-

гается, что ошибочное слово хранится в переменной **error_word**:

```
AspellWordList *suggestions = aspell_speller_suggest (speller, error_word, -1);

if (! suggestions)
   ; //например, выходим по return

//иначе получаем список предположений:

AspellStringEnumeration *elements = aspell_word_list_elements (suggestions);
const char *word;

//и перебирая их (elements) по одному, печатаем в консоль:

while (word = aspell_string_enumeration_next (elements))
       printf (“%s \n”, word);

//удаляем объект:
delete_aspell_string_enumeration (elements);
```

Не буду вдаваться в подробности внутреннего устройства этих функций – скажу лишь, что API можно было сократить наполовину. Память, полученную от aspell_speller_suggest(), освобождать не нужно – возвращается указатель на const. Объект доступен до следующего вызова вышеупомянутой функции.

Вот мы и рассмотрели основные функции *Aspell*. Напоследок расскажу еще об одной, которой программисты пользуются очень часто – это добавление нового слова в словарь. Пример очевиден:

```
aspell_speller_add_to_personal (speller, word, strlen (word));
aspell_speller_save_all_word_lists (speller);
```

Размер слова указывается в байтах, поэтому годится обычная strlen, даже если слово у вас хранится в UTF-8. И не забывайте вызывать aspell_speller_save_all_word_lists(), иначе слово хотя и добавится в текущую сессию проверки орфографии, но не будет сохранено в словарном файле.

ЧАСТЬ 2: Enchant

Enchant – побочный продукт разработчиков *AbiWord* (**www.abisource.com/enchant**). Собственно говоря, это и не движок сам по себе, а программная прослойка, предоставляющая доступ к другим движкам проверки орфграфии, как-то: *Aspell*, *ISpell*, *MySpell*, *Uspell*, *Hspell*, *AppleSpell* и *Hunspell*. Разработчики позаботились об удобном подключении библиотеки при компиляции – то бишь предоставили пакет для *pkg-config*. Поэтому, чтобы проверить наличие библиотеки и подключить ее к своей программе, надо добавить в **configure.in** примерно следующее:

```
echo -n “checking for enchant... “
if pkg-config --exists enchant  ; then
            LIBS=”$LIBS `pkg-config --libs enchant `”
            CFLAGS=”$CFLAGS `pkg-config --cflags enchant `”
            AC_DEFINE(ENCHANT_SUPPORTED, 1, [ENCHANT_SUPPORTED])
            echo “yes”
else
            echo “no”
fi
```

Ну, а в коде программы:

```
#ifdef ENCHANT_SUPPORTED
#include “enchant.h”
#endif
```

Общение с *Enchant* происходит в кодировке UTF-8, вам не нужно указывать это напрямую. Инициализация словаря выполняется просто:

```
EnchantBroker *broker = enchant_broker_init ();
EnchantDict *dict = NULL;

gchar *lang = g_getenv (“LANG”);
if (lang)
   dict = enchant_broker_request_dict (broker, lang);
```

Вначале создается брокер, у которого запрашиваем словарь: брокер ими заведует. В приведенном выше примере мы просим у брокера: «Дай нам словарь для языка текущей локали». Но ведь известно, что *Enchant* поддерживает одновременно несколько движков проверки орфографии. Стало быть, *Enchant* даст нам (прозрачно, разумеется) тот движок, в котором установлен модуль проверки нужного нам языка. Но что если такой модуль есть для нескольких движков? Например, для *Aspell* и *MySpell*? Для этих целей существует файл **/usr/share/Enchant/Enchant.ordering**, в котором задаются приоритеты движков. Расположение этого файла у вас может быть иным – всё зависит от того, где установлен *Enchant*. Может также существовать аналогичный пользовательский файл, хранящийся в **~/.enchant**.

Далее, где именно *Enchant* ищет словари? Смотря какие. Для *MySpell*, *Ispell*, и *Uspell* по умолчанию – в подкаталогах **/usr/share/enchant** (опять же, у вас может быть иначе). Например, словарь *MySpell* ищется в **/usr/share/enchant/myspell**. Что до *Aspell*, то к нему *Enchant* ищет «общие» словари, без всякой привязки к конкретной установке.

Для проверки слова на правильность написания надо использовать следующую функцию:

```
int
enchant_dict_check (EnchantDict *dict, const char *const word, ssize_t len)
```

Она возвращает **0**, если слово присутствует в словаре, положительное значение – если слова там нет, и отрицательное в случае внутренней ошибки движка. Параметры функции: dict – экземпляр словаря, word – передаваемое для проверки слово (в UTF8), len – длина этого слова в байтах, можно использовать strlen либо значение **-1**, если строка завершается нулем.

Получение списка предположительно верных написаний слова:

```
size_t out_n_suggs;
gchar **words = enchant_dict_suggest (dict, s, -1, &out_n_suggs);
```

Здесь слова-предположения помещаются в строковой массив words. Количество элементов массива возвращается в переменной out_n_suggs. А чтобы освободить память, отведенную под этот массив, нужно сделать так:

```
enchant_dict_free_string_list (dict, words);
```

Наконец, после работы с брокером и словарем нужно тоже очищать память:

```
if (dict)
   enchant_broker_free_dict (broker, dict);
if (broker)
   enchant_broker_free (broker);
```

Легко видеть, что API у *Enchant* более простое, чем у *Aspell*. Использование того или иного движка зависит от программы. Мне кажется, что наилучшее решение – это поддержка сразу нескольких движков в зависимости от того, какие из них установлены. Собственно, этим и занимается *Enchant*, но *Aspell* более традиционен для Linux и работает «из коробки» в большинстве дистрибутивов. Кроме того, даже при наличии *Enchant*, *Aspell* с его 70 словарями наверняка будет использоваться тем же *Enchant* в качестве движка. LXF

Программирование со вкусом Ананаса Изучаем учетную платформу со свободной лицензией

Документы и

ЧАСТЬ 4 Завершая цикл статей, посвященных платформе Ананас, **Андрей Паскаль** покажет, как создавать учетные документы и выводить их на печать при содействии *OpenOffice.org*.



Наш эксперт

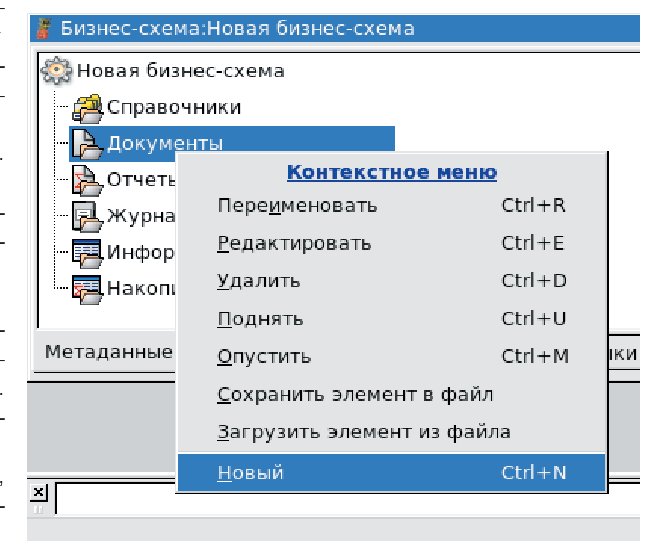
Андрей Паскаль использует Linux более семи лет и играет роль координатора проекта Ананас. Он также знает ответ на популярный у всех новичков вопрос: «Какой дистрибутив Linux мне выбрать?».

Как известно, в автоматизации учетной деятельности и бизнес-процессов принято выделять два методических подхода: «от первичных документов к операциям» и «управление процессами». Программная система, реализующая первый подход, обслуживает потребности пользователя, главным образом, по формированию и учету необходимой документации, как правило, регламентированной. Программные системы, реализующие второй подход, выстраиваются под потребности системы управления организацией, помогая управляющему персоналу держать под контролем качество выполнения бизнес-процессов сотрудниками организации в соответствии с принятым нормами и стандартами. По данной классификации, Ананас относится к системам первого типа, в которых учет строится на регистрации учетных событий в документах. Поэтому разработчику бизнес-схемы необходимо уметь определять документы, их экранные и печатные формы. Учетные документы – это отправная точка, источник данных, на котором строятся все прочие операции, осуществляемые Ананасом.

Как всем известно с детства, «чтобы продать что-нибудь ненужное, нужно сначала купить что-нибудь ненужное», поэтому мы рассмотрим проектирование бизнес-объекта Документ на примере документа Приходная накладная, который используется при постановке на учет приобретенных товаров.

Наша Приходная накладная позволяет хранить номер накладной, дату ее составления, название продавца, выбранного из Справочника контрагентов, а также дополнительную информацию в поле Основание. Все эти реквизиты составляют шапку накладной. Разумеется нам требуется хранить также информацию о приходуемых товарах, их закупочной цене и количестве. Так как каждая накладная может содержать информацию более чем об одном товаре, и их общее количество также может различаться, нам необходима таблица, позволяющая заносить в накладную информацию для требуемого числа позиций.

Первым шагом создадим новый документ, вызвав для этого соответствующий диалог, как показано на **Рис. 1**, и зададим название «Приходная накладная», как показывает **Рис. 2**. Описание вводить не обязательно, однако, это удобное место, чтобы оставить послание тому, кто решит когда-нибудь исправить созданный вами документ. И даже если это будете вы сами, через, скажем, года два, это все равно будет другой человек, так как через такое время вы уже с трудом вспомните суть деталей проекта двухгодичной давности. Если вы сможете сжато и доходчиво объяснить назначение и контекст жизни документа, обязательно сделайте это. Сегодня ничто так не дорожает, как время. Хорошие комментарии экономят его.



› Рис. 1. Вызов диалога создания нового документа.

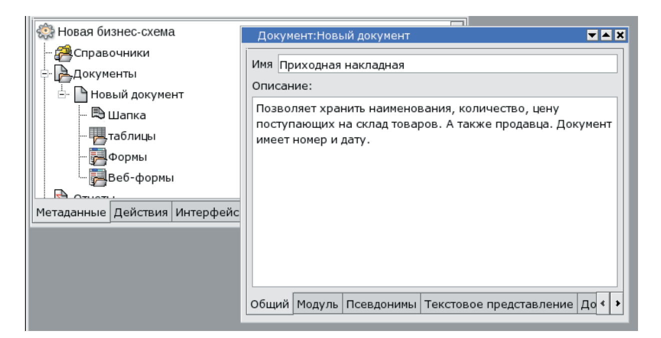
Далее закроем этот диалог и откроем диалог создания атрибута шапки Приходной накладной, как показано на **Рис. 3**. Понятие «атрибут/реквизит шапки документа» подразумевает, что при построении экранной формы, поле редактирования этого реквизита будет находиться где угодно, но только не в табличной части документа. Иными словами, поле для редактирования атрибута шапки на самом деле может находиться в подвале (нижней части) экранной формы.

Определим атрибут Номер, указав тип данных Символ(100), как показывает **Рис. 4**. Закроем диалог и аналогично определим атрибуты Дата, Продавец, Основание, задав типы данных Дата, Справочник контрагентов, Символ соответственно. Получилось? Разумеется, невозможно задать тип данных Справочник контрагентов, пока его еще нет. Это я говорю для тех, кто пропустил предыдущие уроки, где мы как раз и занимались его созданием. Им следует сделать паузу в работе над Приходной накладной и заняться Справочником контрагентов, как объясняется во второй статье цикла (LXF98). Однако, сейчас не обязательно полностью переключаться на Справочник контрагентов, достаточно создать его, указав название. Определение реквизитов контрагентов можно оставить и на потом. После создания Справочника конт-

» Месяц назад Мы размещали меню и панели инструментов на экранной форме Ананаса.

печатные формы

рагентов полю Продавец можно будет, наконец, задать тип Справочник контрагентов, что демонстрирует **Рис.5**.



› Рис. 2. Диалог создания нового документа.

Дела табличные

Задав реквизиты шапки документа, определим состав столбцов таблицы, в которую будет заноситься информация по товарам.

Для каждого товара в накладной нам достаточно хранить его наименование, цену, количество. Также полезно будет хранить сумму и примечания. Значения ячеек столбца Сумма будет вычислять код модуля экранной формы следующего вида

```
/**
 *          Вызывается при изменении значения в ячейке таблицы с
именем tname
 */
function on_tabupdate(row,col,tname)
{
            // только для таблицы wDBTable1. wDBTable1 - имя таблицы,
            // задаваемое в дизайнере, а не имя в метаданных
   if(tname!="wDBTable1")  return;
            // подсчет суммы и запись ее в поле «Сумма» таблицы
   var newValue = parseFloat(TabValue(tname,row,1))*parseFloat(TabValue(tname,r
ow,2));
   SetTabValue(tname,"Сумма", row,          newValue);
}
```

Функция on_tabupdate(row,col,tname), является зарезервированной служебной функцией *Ананаса*. Если разработчик бизнес-схемы определит в модуле экранной формы документа или другого бизнес-объекта функцию с таким именем и с таким количеством параметров (имена параметров могут быть любыми), то она будет вызываться всякий раз, как только пользователь изменит значение любой ячейки таблицы экранной формы. Такие функции принято называть обработчиками событий. Самым очевидным применением обработчика on_tabupdate является расчет вычисляемых значений типа поля Сумма или Сумма по документу.

Важный момент, связанный с табличным представлением данных в документе, который следует осознать до начала использования таблиц – это возможность определения в документе более одной таблицы. *Ананас* не ограничивает количество табличных частей у документа, проектируемого разработчиком бизнес-схемы. Документ может не содержать ни одной табличной части, как например, платежное поручение, или содержать одну, две и более табличных частей. Честно говоря, мой скромный объем знаний состава первичных документов различных участков учета не помогает мне вспомнить хотя бы один вид, где бы потребовалось больше двух табличных частей. Но видимо, разработчики *Ананаса* решили не ограничивать запас страховки и в этом случае.

Таким образом, перед определением первого реквизита табличной части документа необходимо создать эту табличную часть.

Столбец Наименование имеет сложный тип данных, его значение задается выбором из Справочника товаров. Поэтому для задания типа этому столбцу табличной части необходимо уже иметь Справочник товаров в дереве метаданных.

Проектирование структуры документа *Ананаса* очень похоже на проектирование структуры справочника, которая была освоена нами на прошлых уроках на примере Справочника контрагентов. Во многом схоже и создание экранных форм справочника и документа. А вот чем документ действительно отличается от справочника – это наличием печатной формы. Едва ли найдется первичный документ, которому она была бы не нужна, тогда как у справочников как раз наоборот,

› Рис. 3. Создание нового реквизита шапки документа.

› Рис. 4. Задание свойств реквизита Номер документа.

Поле:Продавец

Имя: Продавец

Тип: Справочник.Справочник Контрагентов

Флаг

- Сорт
- Положительное
- Не нулевое

Источник остатков

Общий | Описание | Псевдонимы

Рис. 5. Реквизит Продавец имеет сложный тип данных.

В чем сила, брат?

«Вот в чем сила!» – просто не может не воскликнуть любой программист, впервые увидев, какой мощный арсенал готовых, качественных и, что самое главное, хорошо знакомых пользователю инструментов ему предоставляет мир свободного ПО для реализации его идей в форме потребительских программных продуктов. Ярчайшим примером подобного случая является пакет офисных приложений *OpenOffice.org*, точнее сказать, его доступность. Ни один проприетарный продукт не может рассчитывать в отношениях пользователей к себе на такой уровень лояльности, который имеет *OpenOffice.org* среди пользователей и разработчиков. Создатели Ананаса, входя в их число, смогли максимально воспользоваться всеми преимуществами этого пакета программ, интегрировав Ананас с *OpenOffice*.org в настолько удобной для пользователя форме, что порой он даже не замечает и не задумывается над тем, что подготовленный в Ананасе счет или накладная открываются для печати в редакторе Writer или в электронных таблицах Calc из пакета *OpenOffice.org*. Такая интеграция стала возможной благодаря открытости форматов файлов этого офисного пакета. Формат ODF, используемый программами пакета *OpenOffice.org* является сегодня стандартом, утвержденным ISO, международной организацией по стандартизации.

печатные формы встречаются редко. А если и есть, то называются они обычно отчетами, которые живут в Ананасе самостоятельной жизнью. В общем, как вы уже догадались, мы переходим к рассмотрению такой важной темы для любой учетной системы, как печатные формы и отчеты.



Рис. 6. Реквизиты шапки и табличной части документа определены.

Твердая копия

Подсистема подготовки печатных форм (отчетов) *Ананаса* использует хорошо известную методику с использованием шаблонов. Суть ее заключается в создании готового документа при помощи соединения шаблона документа, задающего форматирование и оформление, и данных, обработанных и подготовленных специальным образом для вписывания в шаблон. Такой подход хорошо себя зарекомендовал тем, что обеспечивает поддержку большого количества различных форматов. Например, *Ананас* с одинаковой легкостью может формировать отчеты в форматах TeX, HTML, XML. Какой формат выбрать, решает разработчик бизнес-схемы. В бизнес-схеме, поставляемой в дистрибутиве *Ананаса*, используется формат документов *OpenOffice.org*, как наиболее дружественный и удобный для пользователя, привыкшего к работе с OOo.

Рассмотрим возможности *Ананаса* по работе с отчетами с точки зрения разработчика бизнес-схемы. С точки зрения пользователя, отчет или печатная форма документа – это просто документ, открытый *Ананасом* либо в редакторе *Writer*, либо в электронной таблице *Calc* пакета *OpenOffice.org*.

Для разработчика бизнес-схемы подготовка к формированию отчета складывается из двух составляющих: подготовки шаблона отчета и написания кода, команды которого формируют отчет из шаблона и данных.

Формирование печатной формы осуществляет *Ананас-Скрипт* на основании имеющихся в его распоряжении данных и шаблона отчета. Шаблон отчета представляет собой документ *OpenOffice Writer*, с занесенными в него тегами специального формата. Процедура на *Ананас-Скрипте* подменяет теги значениями, оставляя неизменным стилевое оформление, заданное для тега. **Рис. 7** показывает шаблон приходной накладной. Каждый тег в документе начинается с пары символов <: и заканчивается парой символов :>, между которыми указывается имя тега. Его-то и использует *Ананас-Скрипт* при формировании отчета.

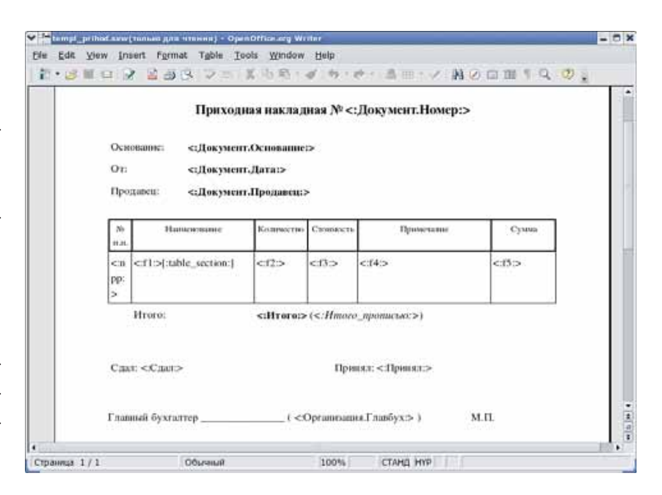


Рис. 7. Вид шаблона печатной формы Приходной накладной.

Приведем готовый код процедуры, осуществляющей формирование печатной формы Приходной накладной.

```
/**
 * обработчик нажатия кнопки
 */
function on_button(button_name)
{
   if(button_name!="print_button") return; // обрабатываем только нажатие кнопки Печать
   var p = new Report("Report1","1"); // создаем экземпляр отчета
   p.setTable("templ_prihod.odt"); // связываем его с шаблоном
   // шаблоны ищутся в каталоге `Рабочий каталог'
   p.setValue("param",Value("Номер") ); // устанавливает значение для простого тега
   p.exec("Документ.Номер"); // применяет установленное значение для тега с именем <:Документ.Номер:>
   p.setValue("param",Value("Основание") );
   p.exec("Документ.Основание");
   p.setValue("param",Value("Продавец"));
   p.exec("Документ.Продавец");
```

```
    p.setValue("param",Value("Дата") );
    p.exec("Документ.Дата");
    var sum=0;
    var countRow =  TabCount("wDBTable1");   // количество строк таблицы
    var i;
    for ( i =countRow-1; i>=0; i--)
    {
                p.setValue("npp", String(i+1)); // устанавливаем значение табличного тега с именем `npp'
                for ( j=1; j<=5; j++)
                {
                        p.setValue("f"+String(j), TabValue("wDBTable1",i,j-1));
                }
                p.exec("table_section"); // применяем установленные значения для строки
                                //содержащей табличный тег [:table_section:]
    }
    sum = Value("lineEdit1");
    p.setValue("param",sum);
    p.exec("Итого");
    p.setValue("param", Propis(sum)); // записываем сумму прописью
    p.exec("Итого_прописью");
    p.setValue("param","Сидоров С.С");
    p.exec("Сдал");
    p.setValue("param","Петров П.П");
    p.exec("Принял");
    p.setValue("param",getConstant("Константы","Главный бухгалтер")); // получаем имя главного бухгалтера из справочника констант
    p.exec("Организация.Главбух");
    p.show(); // запускаем OpenOffice
    p.close(); // удаляем все временные файлы
}
```

Этот код следует поместить в модуль экранной формы Приходной накладной. Он сработает только если:

» Была нажата кнопка, для которой в дизайнере экранных форм было задано имя print_button.

» Шаблон Приходной накладной, показанный на **Рис. 7** хранится в файле **templ_prihod.odt**, расположенном в рабочем каталоге. Рабочий каталог задается в параметрах настройки бизнес-схемы.

» Названия справочников и полей совпадают с указанными в коде.

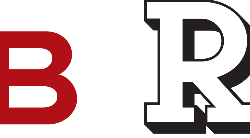
Внимательный читатель уже самостоятельно заметил наличие в шаблоне странного тега с прямоугольными скобками [:table_section:]. Такой тег нужен для таблицы. Так как в шаблоне определяется только одна строка таблицы, а в сформированном документе их может быть произвольное количество, необходим тег, который позволил бы программе понять, что эту часть шаблона следует рассматривать как размножаемую строку. Имя тега можно выбирать произвольно – важно лишь то, чтобы и в шаблоне, и в коде оно было одинаковым.

Более детальное описание возможностей и способов работы подсистемы формирования отчетов *Ананаса* следует искать в фирменном руководстве. Самый быстрый и качественный ответ на практически любой технический вопрос по этой теме всегда можно получить на форуме проекта *Ананас*. Дерзайте! LXF



Новая серия! Свободный инструментарий для статистической обработки данных



Введение в R

ЧАСТЬ 1 Вы ещё анализируете данные по старинке, методом внимательного вглядывания в набор точек на графике, изредка подгоняя их к прямой с помощью многочисленных «мышекликов»? Пусть это делает за вас компьютер и R – это именно тот язык, который поможет доходчиво разъяснить ему вашу проблему.

Наши эксперты



Шипунов Алексей
Биолог, преподаватель, сотрудник американского университета Айдахо, использующий R для обработки информации во всех своих проектах, начиная с 2001 года. Русский переводчик R.



Евгений Балдин
Физик, преподаватель, научный сотрудник ИЯФ им. Будкера, профессионально занимается обработкой экспериментальных данных в области физики высоких энергий. Давний сторонник свободного ПО, впечатлённый мощью R.



Прежде всего, R – язык программирования для статистической обработки данных и работы с графикой, и в то же время – это свободная программная среда с открытым исходным кодом, развиваемая в рамках проекта GNU. R можно найти в любом дистрибутиве, не ставящем своей целью уместиться на одну дискетку. Например, в Debian GNU/Linux базовый пакет носит имя *r-base*, а подключаемые модули проще всего искать по акрониму «cran».

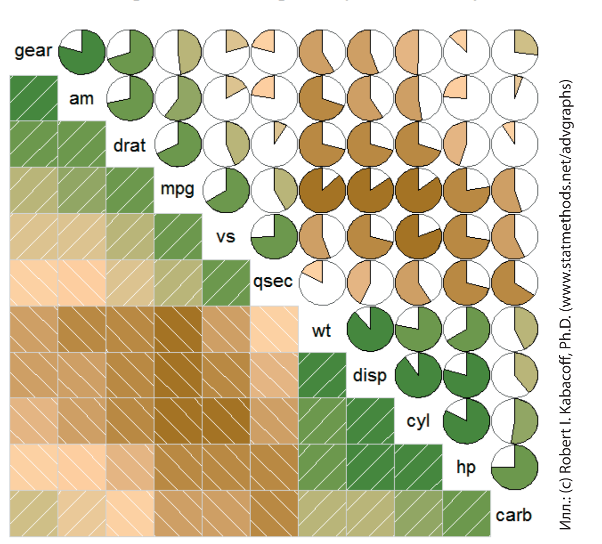
R применяется везде, где нужна работа с данными. Это не только статистика в узком смысле слова, но и первичный анализ (графики, таблицы сопряжённости), и продвинутое математическое моделирование. R без особых проблем может использоваться и там, где сейчас принято использовать коммерческие программы анализа уровня *MatLab*/*Octave*. С другой стороны, вполне естественно, что основная вычислительная мощь R лучше всего его проявляется при статистическом анализе: от вычисления средних величин до вейвлет-преобразований временных рядов.

География использования R очень разнообразна. Трудно найти американский или западноевропейский университет, где бы не работали бы с R. Очень многие серьезные компании (например, Boeing) используют R в своей деятельности. R для статистиков – это действительно глобально.

Немного истории

R возник как свободный аналог среды S-PLUS, которая, в свою очередь, является коммерческой реализацией языка расчётов S.

Язык S – довольно старая разработка (почти как *TeX*). Он возник ещё в 1976 году в компании Bell Labs, и был назван, естественно, «по мотивам» языка С. Первая реализация S была написана на FORTRAN и работала под управлением операционной системы GCOS. В 1980 г. она



Илл.: (c) Robert I. Kabacoff, Ph.D. (www.statmethods.net/advgraphs)

была перенесена в UNIX, и с этого момента S стал распространяться, в основном, в научной среде. Начиная с третьей версии (1988 г.), коммерческая реализация S называется S-PLUS. Последняя в настоящее время продвигается компанией Insightful, и доступна под Windows и различными версиями UNIX, естественно, за плату, причем весьма и весьма немаленькую (для UNIX, например, S-PLUS предлагается за $6500). Собственно говоря, именно высокая цена и сдерживала широкое распространение этого во многих отношениях замечательного продукта. Тут-то и начинается история R.

В августе 1993 г. двое молодых новозеландских ученых анонсировали свою новую разработку, которую они назвали R. По замыслу создателей, Роберта Джентльмена [Robert Gentleman] и Росса Ихака [Ross Ihaka], она должна была стать новой реализацией языка S, отличающейся от S-PLUS некоторыми деталями, например, обращением с глобальными и локальными переменными, а также работой с памятью. Фактически, они создали не полный аналог S-PLUS, а новую «ветку» на «дереве S». Многие вещи, которые отличают R от S-PLUS, связаны с влиянием языка Scheme [функциональный язык программирования, один из наиболее популярных диалектов языка Lisp, – *прим. авт.*].

Сначала проект развивался довольно медленно, но когда в нём появилось достаточно возможностей, в том числе уникальная по лёгкости система написания дополнений или пакетов, всё большее количество людей стало переходить с S-PLUS на R. Когда же, наконец, были устранены свойственные первым версиям проблемы с памятью, то среди пользователей R стали появляться и любители других статистических пакетов (прежде всего тех, которые имеют интерфейс командной строки: *SAS*, *Stata*, *SYSTAT*). Количество книг, написанных про R, за

последние годы выросло в несколько раз, а количество пакетов уже приближается к полутора тысячам.

Идея центральной системы хранения и распространения пакетов, CRAN или Comprehensive R Archive Network (http://cran.r-project.org/), была заимствована из TeX-сообщества (CTAN или Comprehensive TeX Archive Network; аналогичной схемой пользуется и Perl-сообщество: CPAN или Comprehensive Perl Archive Network). Все три упомянутых проекта объединяет одно: стабильная база и множество дополнений. В отличие от добавления новой функциональности в монолитную программу, качественный пакет может сравнительно легко написать один человек за вполне обозримый промежуток времени.

Как скачать и установить R

Поскольку R – свободное ПО, его можно скачать и установить совершенно бесплатно. Конкретный способ, конечно, будет зависеть от установленной у вас ОС. Как уже говорилось выше, R входит в репозитории большинства распространенных дистрибутивов Linux, единственное, что нужно учесть – обновление пакетов часто отстает от выхода официальных версий (релиз-цикл R длится примерно три месяца, на момент написания статьи актуальной была версия 2.6.0). Версия R нумеруется тремя числами: первые два – это главная версия, которая обновляется два раза в год. С каждой главной версией в R привносятся изменения, причём часто – довольно значительные. Как правило, это множество новых команд, улучшенные алгоритмы выполнения старых, и, разумеется, исправления ошибок. К недостаткам смены версии можно отнести возможные проблемы с обратной совместимостью. Естественно, разработчики стараются свести такие изменения к минимуму. С другой стороны, написанные на R программы пяти-семилетней давности, как правило, работают без проблем. В общем и целом, мораль такова: обновляйте R смело, но при этом всегда читайте список изменений.

На каждую главную версию выходит, как правило, две минорных (нулевая и первая). Первая минорная версия обычно ничего нового, кроме исправления ошибок, не привносит. Таким образом, если Вы хотите всегда иметь самую свежую версию, то репозиторий пакетов, особенно в случае стабильных дистрибутивов, не годится. В этом случае надо будет скачивать R из CRAN (http://cran.r-project.org/). У этого сайта довольно много зеркал, так что можно выбрать подходящее.

Процедура компиляции R из исходного кода вполне стандартная: *./configure*, *make* и *make install* от имени суперпользователя. В общем, если процесс компиляции как таковой вас не пугает, собрать R из исходных текстов не составит труда.

Есть одна, важная для всех операционных систем особенность: R (в отличие от того же S-PLUS) держит все свои вычисления в оперативной памяти, поэтому, если в процессе работы, скажем, выключится электропитание, то результаты сессии, не записанные явным образом в файл, пропадут. Эта особенность, к сожалению, не позволяет R работать с действительно большими объёмами данных (порядка сотен тысяч и более записей), отдавая их на откуп гораздо менее удобной системе анализа ROOT (http://root.cern.ch), про которую мы уже говорили в LXF83.

Запуск

Опять-таки, каждая операционная система имеет свои особенности работы. Но, в целом, можно сказать, что и в Linux, и в Mac OS X, и в Windows существует так называемый «терминальный» способ запуска (для Mac и Windows имеется и штатный GUI с некоторыми дополнительными возможностями).

Терминальный способ прост: достаточно набрать в командной строке:

```
=> R

R version 2.4.0 Patched (2006-11-25 r39997)
Copyright (C) 2006 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R -- это свободное ПО, и оно поставляется безо всяких
гарантий. Вы вольны распространять его при соблюдении
некоторых условий. Введите 'license()' для получения
более подробной информации.

R -- это проект, в котором сотрудничает множество
разработчиков. Введите 'contributors()' для получения
дополнительной информации и 'citation()' для ознакомления
с правилами упоминания R и его пакетов в публикациях.

Введите 'demo()' для запуска демонстрационных программ,
'help()' -- для получения справки, 'help.start()' --
для доступа к справке через браузер. Введите 'q()', чтобы
выйти из R.

>
```

и появится приглашение в виде символа >. Теперь можно приступать к работе.

Если терминал запущен вне графической среды, то все изображения будут скидываться в один многостраничный PostScript-файл – **Rplots.ps**. В Mac OS X это будет происходить, даже если X11 запущен, так что полноценно использовать R под Mac можно только в GUI-варианте. Терминальный запуск под Windows таких ограничений не имеет.

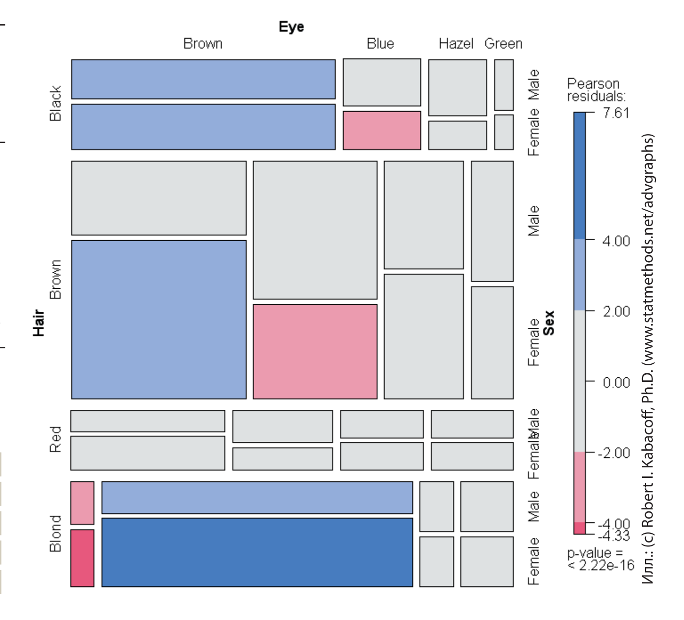
На будущее договоримся, что под «сессией R» мы будем иметь в виду терминальный запуск под X11 в GNU/Linux и GUI-запуск в Windows и Mac. GUI под эти операционные системы построены так, что они всё равно запускают терминал-подобное окно. Общение с R возможно только в режиме диалога «команда-ответ». Полноценного GUI с R не поставляется: хотя предпринимаются многочисленные попытки создать такую систему, но пока что все они далеки от завершения. Возможно, это и к лучшему, так как система из меню-окошек-опций не способна заменить полноценный интерфейс командной строки, особенно в случае таких сложных систем, как R. Интересно, что S-PLUS имеет очень приличный GUI, но если открыть любой учебник по этой системе, то можно заметить, что автор настоятельно рекомендует пользоваться командной строкой.

Первые шаги

Перед тем как начать работать, надо понять, как выйти. Для этого достаточно ввести одну команду и ответить на один вопрос:

```
> q()
> Save workspace image? [y/n/c]: n
```

Уже такой простой пример демонстрирует, что любая команда в R – это функция, которой можно передать аргумент. Даже если



Илл.: (c) Robert I. Kabacoff, Ph.D. (www.statmethods.net/advgraphs)

аргумент не указан, то скобки всё равно надо ввести. Если этого не сделать, то вместо выхода из R на экран будет выведено определение функции:

```
> q
 function (save = "default", status = 0, runLast = TRUE)
.Internal(quit(save, status, runLast))
<environment: namespace:base>
```

Чтобы узнать, как правильно вызывать функцию, следует воспользоваться встроенной справкой. Есть два пути. Первый – вызвать команду справки:

```
> help(q)
```

или

```
> ?q
```

Текст справки будет выведен в основном окне программы. Если внимательно прочитать его, то станет ясно, что выйти из R можно, и не отвечая на дополнительный вопрос, если ввести:

```
> q("no")
```

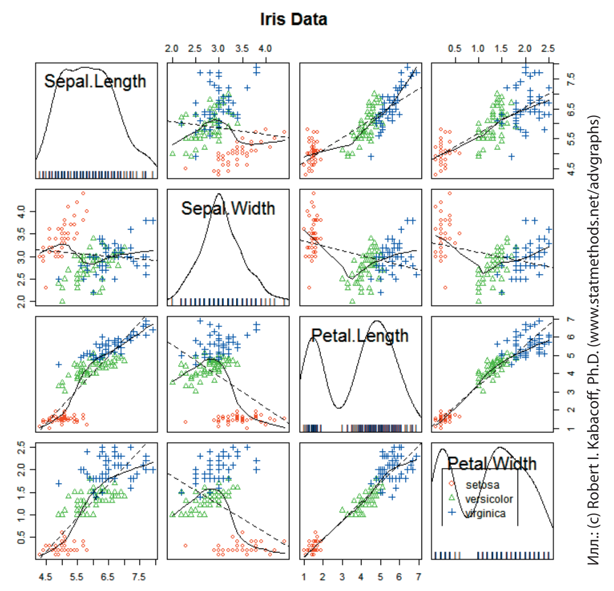
Зачем же нужен этот вопрос? Или, другими словами, что будет, если ответить положительно? В этом случае в рабочую папку R (ту, из которой он вызван), запишутся два файла: бинарный **.RData** и текстовый **.Rhistory**. Первый содержит все объекты, созданные за время сессии. Второй – полную историю введённых команд. R работает с историей команд стандартным образом: доступ к предыдущей команде осуществляется через клавишу-стрелку «вверх».

Если при выходе сохранить файл **.Rhistory**, то команды из этой сессии будут доступны и в следующей, при условии, что R будет вызван из того же самого каталога. И наоборот, если случайно сохранить рабочую среду (эти два файла), то при следующем старте они загрузятся автоматически. Иногда такое поведение R становится причиной различных недоумений, так что необходимо быть внимательным.

Итак, как войти и как выйти, уже понятно. Осталось сказать ещё немного про помощь. Её можно вызвать несколькими различными способами. Во-первых, посредством команд ? или help(), как описано выше. Во-вторых, можно вызвать команду help.start(). В этом случае откроется окно браузера, в котором будет демонстрироваться так называемая HTML-помощь. Основное её преимущество перед обычной текстовой в том, что разделы соединены гиперссылками (в Mac OS X такая помощь вызывается обычными командами). В-третьих, вместе с R устанавливается несколько руководств в формате PDF. Их можно найти в папке, содержащей документацию.

Наконец, часто бывает нужна «обратная» помощь – вы знаете, что вы хотите, но не знаете, как это сделать (какую команду вызвать). В этом случае могут помочь две команды: help.search() и apropos(). Вот как их надо вызывать:

```
> help.search("vector")
```



Илл.: (c) Robert I. Kabacoff, Ph.D. (www.statmethods.net/advgraphs)

Результатом выполнения будет список команд с кратким описанием их действия. Команда apropos() выдает простой список команд, содержащих строку, которая была указана в кавычках. Кстати, обратите внимание – кавычки следует использовать обязательно. Их можно использовать и в обычных командах помощи, например, help("q") или ?"q", причём иногда эти команды без кавычек просто не работают (например, нельзя получить справку по символу плюса, введя ?+, надо использовать ?"+"). Если ничего не помогает, и найти нужную функцию не удается, то приходится обращаться за справкой в Интернет или в список рассылки R-help.

Калькулятор-переросток

Так назвал один из вводных разделов своей книги «Introduction to R» один из создателей системы, Питер Далгаард [Peter Dalgaard]. Это значит, что R можно использовать, в том числе, и как обычный калькулятор. Например:

```
> 3+2
[1] 5
> 16+3/5-11*8^2
[1] -687.4
> ((((16+3)/5)-11)*8)^2
[1] 3317.76
```

Для знакомых с интерактивными языками программирования типа Python здесь нет ничего необычного. Единственная любопытная деталь – это единичка в квадратных скобках. Она означает номер элемента вектора. Отсюда сразу два логических вывода:

1. R трактует результат любой операции с числами как вектор единичной длины. Скаляров в R, вообще говоря, нет;
2. Элементы векторов нумеруются с единицы, а не с нуля, как принято во многих языках программирования.

Для читателей, менее знакомых с программированием, отметим, что порядок арифметических действий в R стандартный, знакомый со школьной математики. Скобки (раскрывающиеся изнутри наружу) позволяют этот порядок изменять:

```
> # Первый пример
> 3/7
[1] 0.4285714
> 3/7-0.4285714
[1] 2.857143e-08
> # Второй пример
> sqrt(2)*sqrt(2)
[1] 2
> (sqrt(2)*sqrt(2))-2
[1] 4.440892e-16
```

Эти примеры посложнее; кроме того, в них есть подводные камни. Почему разность **3/7-0.4285714** не равна нулю, должно быть понятно, всем, знающим арифметику. При выводе на консоль, R неявно использует функцию print(), которая округляет бесконечную периодическую дробь **0,428571**. Второй пример интереснее. Произведение двух квадратных корней из двойки должно давать **2**, что и происходит. Однако если вычесть из результата **2**, получится какое-то очень маленькое число ($4,440892 \cdot 10^{-16}$). Это происходит потому, что вычисления выполняются на компьютере, который только притворяется, что работает с дробями, в то время как на самом деле оперирует только с целыми числами. Любая компьютерная система расчётов работает подобным образом, и с этим можно только смириться. Ещё один момент: приведённые примеры показывают, как можно пользоваться символом комментария (#).

Скажем ещё немного о работе с аргументами на примере команды round() (округлить). Она принимает два аргумента: число, которое нужно округлить, и значение digits, сообщающее, до какого знака округлять. Система аргументов работает разумно, так что все равно, как именно написать команду:

```
> round(1.5, digits=0)
[1] 2
> round(1.5, d=0)
```

```
[1] 2
> round(d=0, 1.5)
[1] 2
> round(1.5, 0)
[1] 2
> round(1.5,)
[1] 2
> round(1.5)
[1] 2
```

Так происходит благодаря тому, что есть значения аргументов по умолчанию. Например, в данном случае значение по умолчанию для аргумента digits – **0**. Об этом говорит и результат вывода команды args():

```
> args(round)
> function (x, digits = 0)
```

Можно также заметить, что некоторые аргументы имеют имена. Значит, аргументы можно задавать не по порядку, а по именам. Имена можно сокращать вплоть до одной буквы, но только если нет разных аргументов, которые от такого сокращения станут неразличимыми. Можно также перечислять аргументы по порядку, через запятую (не забудьте, что для десятичных дробей используется точка!), тогда имена можно опускать.

Скрипты

Просто открыть сессию R и вводить в окно программы команды, одну за другой – это лишь один из возможных способов работы. Гораздо более продуктивный метод, который является заодно и серьёзнейшим преимуществом R – это создание скриптов, иначе говоря – программ, которые потом загружаются в R и интерпретируются им. С самого начала работы следует создавать скрипты, даже для таких задач, которые кажутся пустяковыми – в будущем это значительно сэкономит ваше бесценное время. Создание скриптов по любому поводу и даже без особого повода – одна из основ культуры работы в R.

Создать скрипт очень просто. Допустим, после открытия сессии была введена необходимая для получения искомого результата последовательность команд. Чтобы сделать эту работу воспроизводимой, надо просто сохранить историю. Лучше всего это делать с помощью команды:

```
> savehistory(file="myscript.r")
```

После этого в текущем каталоге будет создан файл **myscript.r**, который легко отредактировать в любимом текстовом редакторе, а потом загрузить обратно в R командой:

```
> source("myscript.r", echo=TRUE)
```

Опция echo добавлена для того, чтобы можно было видеть сами команды, а не только результат их выполнения.

Илл.: (c) Robert I. Kabacoff, Ph.D. (www.statmethods.net/advgraphs)

Есть ещё более эффективный способ работы: вы открываете скрипт в текстовом редакторе, а потом посылаете отдельные его строки прямо в R. Есть несколько редакторов, которые умеют так делать. Во-первых, это *Emacs* с установленным пакетом *ESS* (*Emacs Speaks Statistics*). Прелесть этой системы в том, что R запускается прямо в одном из окон редактора. Во-вторых, штатные R GUI под Windows и под Mac также имеют встроенные редакторы скриптов. Правда, Windows-редактор не подсвечивает синтаксис, и вообще довольно неудобен. Вместо него тем пользователям, которых пугает перспектива освоения *Emacs*, можно порекомендовать специализированный редактор *Tinn-R* (www.sciviews.org/Tinn-R/).

Скрипт R можно выполнить, и не запуская интерактивную сессию. Для этого используются специальные опции командной строки. Например, можно поступить вот так:

```
=> R --no-save < myscript.r > out
```

Опция --no-save предписывает R не сохранять результаты сессии в файлах **.RData/.Rhistory** (фактически, отвечает «no» на упомянутый в начале статьи заключительный вопрос).

Пакеты

Ещё одно важное преимущество R – наличие для него многочисленных расширений или пакетов буквально на все случаи жизни.

Несколько пакетов присутствуют сразу после установки R на компьютер – это так называемые базовые пакеты, без которых система просто не работает (скажем, пакет, который так и называется base, или пакет *grDevices*, который управляет выводом графиков), а также «рекомендованные» пакеты (пакет для специализированного кластерного анализа *cluster*, пакет для анализа нелинейных моделей *nlme* и другие).

Кроме того, можно поставить любой из почти полутора тысяч (!) доступных на CRAN пакетов. При наличии доступа в Интернет, это можно сделать прямо из R командой install.packages() (в Mac и Windows есть соответствующие пункты меню). Если соединение с сетью похуже, то можно скачать исходные тексты (под GNU/Linux) или скомпилированные пакеты (под Mac или Windows) и установить их прямо с диска. Поскольку многие пакеты написаны на FORTRAN или C, перед использованием их необходимо скомпилировать. Для этого существует специальная форма вызова R:

```
=> R CMD INSTALL package.tar.gz
```

Естественно, это всё следует делать, если R устанавливается не из стандартного репозитория дистрибутива GNU/Linux. В противном случае можно поискать пакеты по сочетанию "cran". В Debian GNU/Linux Etch таких ровно 85 – это, конечно, не 1500 пакетов с CRAN, но скорее всего, там уже есть многое из того, что вам необходимо.

Пакет готов к работе сразу после установки – нужно только инициализировать его перед употреблением. Для этого служит команда library().

Полезные ссылки

В заключение хотелось бы представить список самых полезных, на наш взгляд, сетевых ресурсов по R:

- http://www.r-project.org/ – сайт проекта
- http://cran.r-project.org/ – CRAN
- https://stat.ethz.ch/pipermail/r-help/ – список рассылки R-help
- http://finzi.psych.upenn.edu/nmz.html – поиск в материалах по R
- http://www.statmethods.net/index.html – хороший справочный ресурс
- http://zoonek2.free.fr/UNIX/48_R/all.html – ещё один справочный ресурс
- http://pj.freefaculty.org/R/Rtips.html – советы по использованию R

Пожалуй, для первого раза достаточно. LXF

Через месяц Мы узнаем, как подготавливать данные и строить по ним графики.

И·Г·Р·О·С·Т·Р·О·Й





Все о разработке компьютерных игр с использованием свободного программного обеспечения











АНДРЕЙ ПРАХОВ
Участник нескольких игровых проектов, представитель СМИ и начальник www.linuxmedia.ru

Смело, товарищи, в ногу...

Когда-то бытовало мнение, что программирование – это удел одиночек и вообще нечто сродни шаманству. А если при этом писалась игра, то ее создатель приравнивался чуть ли не к Господу Богу. Сейчас, когда искусство игротворения поставлено на конвейер производства, и гиганты типа «EA Games» выбрасывают на рынок по десятку крупных тайтлов в год, подобное явление стоит рассматривать лишь как один из многих видов обычной коммерческой деятельности.

Время идет, игры усложняются, увеличивается количество специалистов, занятых в данной сфере и, казалось бы, одиночке тут делать нечего. А вот и нет, реальность преподносит нам сюрпризы в виде гениальных произведений искусства (иначе и не скажешь), созданных одним человеком или очень небольшой группой энтузиастов. При ближайшем рассмотрении оказывается, что дело не только в хорошей организованности создателей игры, но и максимальной продуманности технической стороны. Ну в самом деле, зачем писать свой редактор уровней, если имеется уже готовый! А если найдется полный комплекс, специально созданный для программирования игр? Сегодня мы как раз рассмотрим один такой...

info@linuxmedia.ru

Новая серия!
Программируем трехмерную игру, набирая минимум кода.

Ни строчки кода!



ЧАСТЬ 1 Наслышаны о том, что создание игр – это тяжелый труд? Так оно и есть – но виртуоз *Blender* **Андрей Прахов** знает обходной путь!

Большинство поклонников открытых систем хорошо знакомы с прекрасной программой для 3D-моделирования и анимации – *Blender* (www.blender3d.org). Но мало кто знает, что *Blender* – это еще и игровой конструктор со встроенным движком. Обладая мощными возможностями по моделированию в совокупности со средой программирования игр, *Blender* позволяет создавать вполне современные по визуальным возможностям приложения. Помимо графической составляющей, редактор предоставляет разработчику возможность программирования на высокоуровневом языке – Python, интеграцию с физическими библиотеками *Bullet* и *Sumo*, создание двоичного файла для разных операционных систем, а также режим непосредственного кодирования процесса «одной мышью». Заинтересовались?

На протяжении четырех уроков мы с вами пройдем все этапы разработки игры, не прибегая к программированию в полном смысле этого слова. Оказывается, вполне возможно создать нечто работающее и в режиме «выдели и щелкни». Конечно, в серьезных, масштабных проектах без знания Python и его реализации в *Blender* не обойтись, но мы оставим эту тему для будущих уроков профессиональной серии.

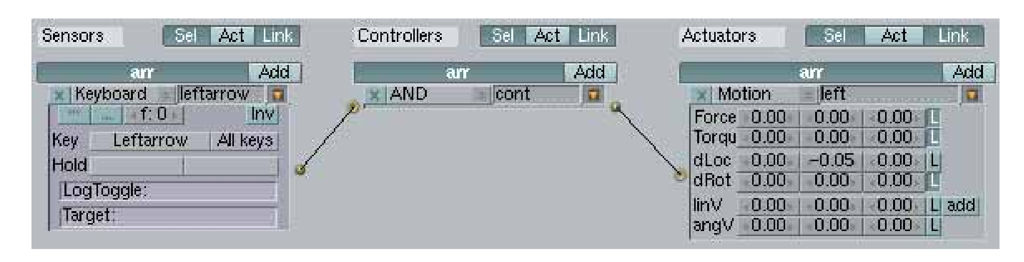
Прежде чем приступить к непосредственному рассмотрению темы урока, советую положить рядом с собой подшивку номеров LXF87/88-LXF91 (ищите их на **LXFDVD**), где рассматривались основы работы с *Blender* и установить последнюю версию программы (на момент написания статьи это 2.45). Можно, конечно, использовать и более старые сборки, однако велика вероятность, что некоторые интерфейсные элементы конструктора не будут совпадать. Уж очень любят разработчики тасовать данные элементы от версии к версии, как карты в колоде! Кроме того, последняя версия *Blender* отличается завидным быстродействием получаемых в итоге исполняемых файлов с игрой.

В первом выпуске «Игростроя» я говорил о пользе и, мало того, о необходимости разработки дизайн-документа для каждого проекта. К счастью, наша игра предполагает быть достаточно простой, поэтому ограничусь перечислением задач и особенностей проекта – этакий примитивный концепт-план.

Проект, над которым мы с вами будем работать, является простым имитатором игры в боулинг. Ради упрощения, правила игры будут изменены:

1. Количество кеглей осталось без изменения, т.е. 10 штук;
2. Количество попыток бросания шара ограничено тремя;
3. Игра имеет начальное меню с двумя пунктами: «start», «end game»;
4. Управление в игре: курсорные клавиши – передвижение указателя направления броска; Пробел – бросок; Esc – выход в меню;
5. Перемещение указателя возможно вправо и влево, сила броска – постоянная;
6. Визуальный интерфейс игры включает в себя отображение всех кеглей, которые исчезают при попадании, и трех шаров, демонстрирующих количество имеющихся у игрока попыток;
7. При выигрыше или проигрыше выводится соответствующая надпись, и по нажатию «any key» происходит возврат в меню.

Вот и все. Просто? Просто, увы, только на бумаге. На протяжении всей серии мы с вами познакомимся с построением игровой логики, реализацией физических законов, правилами игрового текстурирования, с



› Рис. 1. Логическая цепочка для стрелки.

некоторыми интересными визуальными эффектами. Приступим к делу!

Расставляем кегли

Откройте файл **lesson1_begin** – его можно найти в разделе **Журнал/Blender** на **LXFDVD**. Полюбуйтесь появившейся картинкой… и переходите к теории.

Основы визуального программирования заложены в так называемых логических кирпичиках, которые вы сможете увидеть, если нажмете F4. Специфику работы можно легко понять, проведя аналогию с живым организмом. Существует некий раздражитель на который реагируют наши органы чувств, затем идет анализирование полученной информации и подается команда выполнить то или иное действие. В качестве «органов чувств» в *Blender* выступает закладка Sensors [Сенсоры], решение принимается логическим блоком Controllers [Контроллеры], а само выполнение лежит на Активаторах [Actuators]. После выбора нужных кирпичиков (это можно сделать, нажав кнопку Add в каждой закладке) и настройки параметров, необходимо соединить их между собой. Возле каждого кирпичика находится маленькая шашечка. Просто ухватитесь за нее и тяните появившуюся линию к следующему объекту. Удалить соединение можно, указав курсором на нужную линию (при этом она должна выделиться другим цветом) и затем нажав клавишу x. Естественно, любой объект в *Blender* может иметь свою логическую цепочку.

Возвращаемся к уже, надеюсь, открытому проекту. Как было сказано чуть выше, именно стрелка в игре указывает направление производимого броска, а управление осуществляется с помощью соответствующих курсорных клавиш. Итак, выделите объект arr (стрелка), нажмите F4, если вы еще не находитесь в режиме Логика [Logic]. Появившееся окно можно условно разделить на четыре зоны (слева направо): управление физикой объекта и его переменными плюс три логических блока (sensors, controllers, actuators). Пока нас не интересует самый первый блок, физикой мы займемся чуть позже, а вот во всех оставшихся нажмите кнопку Add, тем самым создав для каждой области по новому кирпичику.

Обратите внимание на поля с надписями Always [Всегда], And [И] и Motion [Движение]. Именно здесь вы можете указать конкретную функцию текущего блока. Просто щелкните на них и выберите нужный пункт из меню. Так, для сенсора замените Always на Keyboard [Клавиатура]. Поля рядом позволят вам переименовать эти кирпичики – лучше давать осмысленные названия. Используйте только латиницу и помните о чувствительности *Blender* к регистру букв. Дело в том, что это повышает не только информативность, но и дает возможность обращаться к кирпичикам из Pyhton. Кнопки с нарисованными треугольниками закрывают и открывают поля для редактирования.

Итак, наш сенсор сейчас настроен на работу с клавиатурой. Щелкните мышкой по полю рядом с надписью Key [Клавиша] – должна появится надпись Press a key, нажмите курсорную клавишу Left [Влево]. Теперь, если в игре будет нажата эта клавиша, сенсор тут же выдаст импульс.

Блок Controllers ответственен за принятие решения. Оставьте имеющееся там значение по умолчанию, а вот Motion рассмотрим поподробнее.

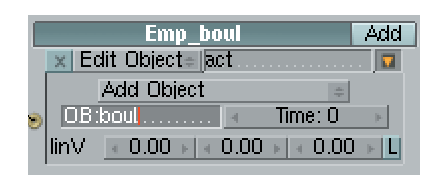
Данный кирпичик позволяет изменять местоположение и векторы направления движения, задавать повороты объекта. Поэтому все данные расположены в трех столбиках, по имеющимся осям XYZ. Первые два пункта, Force [Сила] и Torque [Момент], отвечают за перемещение объекта, если он использует физический движок. Для стрелки это не актуально, поэтому обратите внимание на dLoc и dRot. По названию понятно, что за что отвечает. Введите значение **-0.05** для dLoc по оси Y (средняя колонка) и отожмите справа кнопку L для данного поля, тем самым указав, что нам необходимо работать с глобальными координатами.

Соедините шашечки между всеми тремя блоками для завершения логической цепочки (**Рис. 1**). Чтобы просмотреть результат нашего тяжкого труда, просто нажмите клавишу P, тем самым запустив игровой движок (да, да – *Blender* позволяет полноценно работать с полученной игрой без предварительной компиляции). Теперь, если вы все правильно сделали, нажатие левой курсорной клавиши заставит стрелку послушно ползти влево. Для выхода из режима игры просто нажмите Escape.

Добавьте самостоятельно логику для движения стрелки в другом направлении, а я перехожу к работе с физическим движком. Заставим шар сбивать кегли!



› Рис. 2. Параметры объекта с точки зрения движка Blender.



› Рис. 3. Легкий способ визуализации копии объекта.

Страйк!

Работать с физикой в *Blender* – просто одно удовольствие. Для придания сфере физических свойств, выделите ее и в окне Логика [Logic] нажмите кнопку Actor [Действующий объект] на панели слева. Откроются еще две дополнительные кнопки: Ghost [Невидимость] и Dynamic [Динамика]. Активируйте кнопку Dynamic.

Появившиеся параметры стоит рассмотреть повнимательнее **(рис. 2)**. Поле Mass [Масса] никаких вопросов вызывать не должно, а вот Radius [Радиус] указывает на область чувствительности объекта (отсчитывается от его центра) к столкновениям. Дело в том, что по умолчанию *Blender* создает для каждого физического объекта сферу, которая и принимается в расчет при вычислениях коллизий. Платой за высокую скорость работы служит низкая точность вычислений. К счастью, редактор имеет еще несколько заготовок, в том числе и точный расчет коллизий по всем полигонам объекта. Выбрать подходящий режим можно, нажав кнопку Bounds [Границы]. Из выпадающего списка выберите Convex Hull Polytope [Выпуклый многоугольник].

Ради любопытства можете запустить игру и полюбоваться на падающий шар. Однако настройка его на этом не закончена. Для более реалистичного движения… стоп, у нас и самого движения пока что нет! Давайте исправим этот недочет.

Самые наблюдательные, наверное, уже заметили, что падение шара начинается сразу же с момента запуска программы. Нам же необходимо совершать это действие по нажатию определенной клавиши. Особенностью Blender Engine является то, что отследить момент запуска физики вручную не представляется возможным, поэтому прибегнем к некоторым ухищрениям.

Перенесите сферу в любой другой слой, нажав клавишу M и выбрав соответствующую ячейку. Обратите внимание на dummy-объект Emp_boul [главный слой]. Именно к нему мы привяжем наш код. Задача его очень простая – по нажатию клавиши Пробел создать объект boul, вследствие чего он должен появится на экране и… совершить падение (пока что без движения).

Выделите Emp_boul и создайте по одному кирпичику в каждой области кода. Для Sensors выберите пункт Keyboard и настройте его на реагирование на клавишу Пробел, точно так же, как вы делали это для стрелки чуть выше. В качестве Активатора у нас сейчас выступит кирпичик Edit Object [Правка объекта]. По умолчанию там уже активна функция Add Object [Добавить объект], осталось вписать название создаваемого объекта, т.е. boul, в поле OB: (**Рис. 3**). Соедините шашечки между группами и проверьте результат: на этот раз мы можем произвольно генерировать неисчислимое множество копий шара по нажатию пробела.

И вот теперь настало время придать нашему шарику движение. Для этого познакомимся с еще одним сенсором – Always. Его задача состоит в постоянной генерации импульса, и использовать его можно, скажем, для опроса состояния объекта. В нашем случае, мы настроим сферу на автоматическое движение вперед к кеглям, но работать это будет только после создания самого объекта кодом dummy.

Создайте, как обычно, стандартные кирпичики для сферы и соедините их между собой. Так как сфера



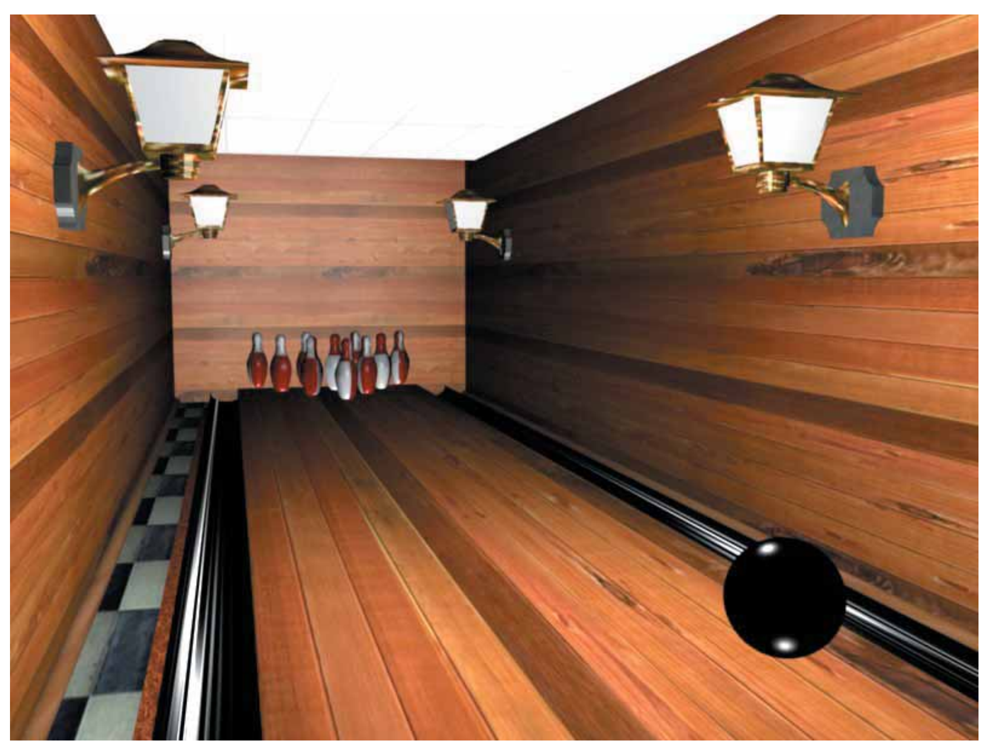
› **Рис. 4. Заставьте кегли чувствовать шарик.**

работает с применением физического движка, то для перемещения ее по одной из осей необходимо использовать параметр Force кирпичика Motion. Поместите в первую ячейку значение **-150** и отожмите кнопку справа, выбрав использование глобальных координат.

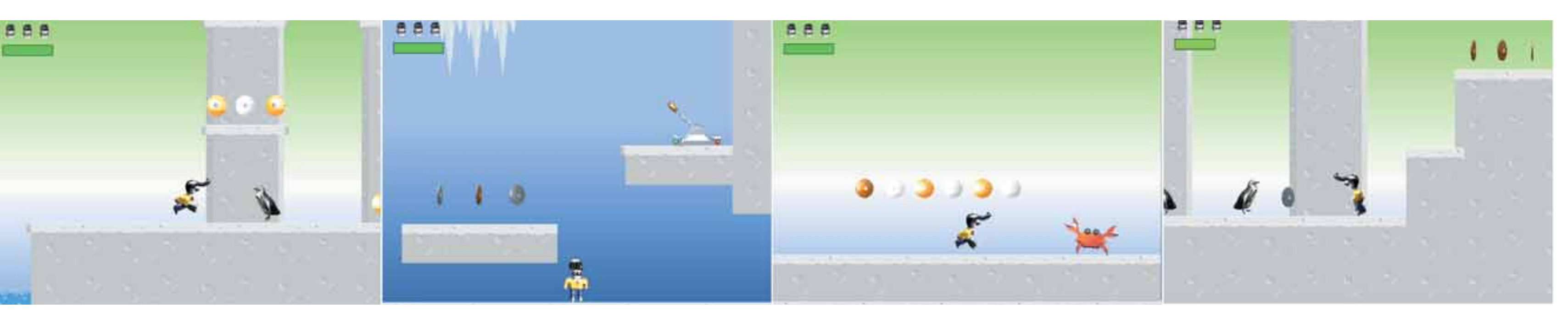
Теперь при нажатии пробела шар послушно заскользит вперед. Для более естественного подобия движения необходимо заставить сферу вращаться при качении. Для этого всего лишь необходимо активировать кнопочку Rigid Body [Твердое тело] в окне Actor. Тем самым, включаются продвинутые возможности имитации физического движения – естественно, за счет дополнительной нагрузки на процессор.

Вот только кегли у нас при этом никак не реагируют на столкновения. Исправить это можно, в свою очередь, подключив их к физическому движку. Для этого настройте каждую из кеглей в соответствии с **рис. 4**.

Ничего сложного, не правда ли? Всего за полчаса мы создали прототип работающей игры, пусть без текстуры, без меню, без системы подсчета и вывода результата, но это работает! LXF



» **В следующий раз** Мы сделаем игру еще увлекательнее, заставив шар следовать за указателем.



Стрелялка за выходные



ЧАСТЬ 2 Сегодня **Александр Супрунов** расскажет о столкновении объектов, анимации, звуке фоновом и нефоновом – в общем, обо всем том, без чего немыслима красивая игра. Время завершать начатое!

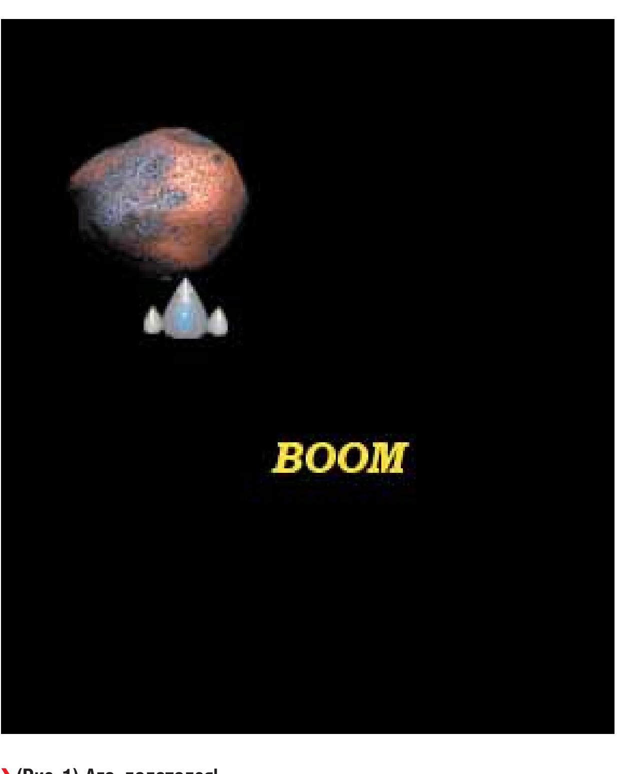
Летел корабль в пустоте. Вдруг – трах, бах!!! Шальной метеорит врезался в борт. Вот это безобразие и называется коллизией. А представьте, если бы ее не было! И метеорит бы мимо пролетел, и... Впрочем, и мы бы провалились сквозь землю. И если в реальной жизни коллизии (столкновения объектов) происходят сплошь и рядом, то и в играх о них нельзя забывать, а значит, потребуется написать хорошую функцию, способную их отследить. Конечно, такая функция в нашей библиотеке уже есть – называется она box(). Вовсе, кстати, не потому, что похожа на одноименный вид спорта (и Патрик, хоть он и..., здесь тоже ни при чем), а из-за метода определения столкновения: берется квадрат (бокс) спрайта и проверяется, не пересекается ли он с таким же квадратом другого спрайта. Данная процедура называется «проверка по боксу». Еще бывает попиксельная, но это уж на крайний случай – слишком много она потребляет ресурсов. Синтаксис вызова box() таков:

box(номер первого спрайта, x-координата первого спрайта, y-координата первого спрайта, номер второго спрайта, x-координата второго спрайта, y-координата второго спрайта);

Функция возвращает **1**, если столкновение произошло и **0** – в противном случае. Вот как можно применить ее на практике:

```
#include "ingame.h"
int main(int n, char **s)
{
  int asteroid_x=100;
  int asteroid_y=150;
  x=250; y=650;
  screen(500, 700);
  loadsprite(1, "ship.bmp");
  loadsprite(2, "asteroid.bmp");
  while (GAME) {
    sprite(1,x,y);
    sprite(2, asteroid_x, asteroid_y);
    if (LEFT){x=x-2;}
    if (RIGHT){x=x+2;}
    if (UP){y=y-2;}
    if (DOWN){y=y+2;}
    if (box (1, x, y, 2, asteroid_x, asteroid_y)) {
      print ("BOOM",200,300);
    }
    fx();
  }
  return 0;
}
```

Здесь мы устанавливаем разрешение 500x700 (как вы, надеюсь, помните, вызов screen() должен быть первым в вашей программе), загружаем спрайты **ship.bmp** (корабль) и **asteroid.bmp** (астероид) в слоты 1 и 2, соответственно, затем, в цикле, отрисовываем их на экране, позволяя изменять координаты корабля клавишами управления курсором, и печатаем громкое "BOOM", если столкновение имеет место. Выход из игры, как и раньше, происходит по нажатию Esc. Напомню, что переменные x и y – встроенные, определять их заново не нужно. Результат работы программы можно видеть на рисунке.



› **(Рис. 1) Ага, долетался!**



Обратите внимание, что вы можете выводить на экран надписи и цифры с помощью функции **print()**.

» **Месяц назад** Мы рассмотрели кирпичики, из которых строится любая стрелялка.

Скорая помощь

Если вам понадобится вывести на экране несколько одинаковых спрайтов, не вздумайте загружать их в разные слоты! Функция **sprite()** вполне справляется с отображением одной и той же картинки в точках с разными координатами.

Если вы пытались уклониться от астероида чересчур активно (он, конечно, неподвижен, но, как утверждал сперва Галилей, а потом Эйнштейн, все относительно), то не могли не заметить, что наша игра обладает одним маленьким недостатком – корабль легко исчезает за пределы видимости. Так происходит потому, что со временем значения координат x и y становится слишком большими и выходят за рамки отведенных 500x700 пикселей. Чтобы исправить ошибку, можно, например, добавить сразу после блока инструкций if следующий код:

```
if (x<=0){x=0;}
if (x>=500){x=500;}
if (y<=0){y=0;}
if (y>=700){y=700;}
```

Происки врагов

Инопланетные захватчики – это не мирные астероиды, и у кораблей, которые будут нас атаковать, возможны различные траектории движения. Какие именно – зависит от вас. Например, большой бомбардировщик может медленно перемещаться по диагонали экрана, методично сбрасывая смертельно опасные, но неповоротливые бомбы, а легкие истребители будут выделывать петли. Иными словами, траектории движения удобно связать с типом противника.

Здесь, конечно, не помешает учебник математики, но особо усердствовать не стоит: если в знаменитой *Galaga* траектории достаточно сложны, это еще не значит, что существует прямая зависимость между их хитроумностью и увлекательностью игры. Попробуйте, для начала, ограничиться теми элементами, которые будут отвлекать вас на дополнительные действия. Например: вы увидели вражеский корабль и начинаете палить по нему из всех орудий. Если бы все было так просто, то секунду спустя его обломки уже затерялись бы в безбрежных просторах мирового эфира. Но вражеский корабль – крепкий орешек: он тоже делает выстрелы, и вам приходится уворачиваться от летящих снарядов, а тут еще этот шальной метеор, вынырнувший неизвестно откуда – и вот вам Game Over, которого никто не ждал. А сколько я там очков набрал? 4100? О, уже больше, чем в прошлый раз. А смогу больше? Собственно, это и называется «увлекательный геймплэй».

Приведем несколько стандартных траекторий движения вражеских объектов для игр, подобных нашей.

1 Сверху вниз по прямой:

```
        sprite(10, evil_x, evil_y);
        evil_y++;
```

2 По диагонали слева направо (или наоборот – замените в последней строке знак «плюс» на «минус»).

```
        sprite(10, evil_x, evil_y);
        evil_y++; evil_x++;
```

Попробуйте также изменять приращения evil_y и evil_x (например: evil_y+=3;). Интересной траектории можно добиться, добавив сюда фактор случайности (evil_x+=rand() %3;), а также внезапную смену направления движения.

3 Движение по кругу. Это более сложный случай, но, в целом, формулы для координат имеют следующий вид: x=r*cos(a)+x0, y=r*sin(a)+y0, где r – радиус окружности, a – угол в радианах. Для использования тригонометрических функций (cos(), sin()) необходимо подключить заголовочный файл **math.h** и задействовать библиотеку *libm* (добавьте *-lm* к командной строке *gcc*).

Попробуйте угадать, как будет двигаться объект в этом примере:

```
float       evil_x=300;
float       evil_y=0;
float radius=150;
float a;
while (GAME) {
  sprite(1, radius*sin(a)+evil_x,radius*cos(a)+evil_y);
  a=a+0.01;
  if (a>=3.14){a=0.;}
  evil_y+=0.5;
  fx();
}
```

Вражеский пилот явно ас: его корабль движется практически случайным образом. Да, поразить такую цель будет непросто... А если вас смущает загадочное число 3,14 – то это просто примерное значение «пи» или число радиан, соответствующее 180 градусам. Обратите внимание на то, что увеличивать координаты можно не только на 1, 2 и так далее, но и на дробные числа – например на 0,5, как в этом примере.

Что за штука... велосити?

Велосити (англ. velocity – скорость) – вероятно, один из самых чудесных способов сделать управление более реалистичным. Что мы имеем сейчас? Нажимаем клавишу – корабль движется, отпускаем – останавливается. Но в реальном мире тела обладают инерционностью, и, как с детства учат нас правила безопасного поведения на дорогах, ни одна машина не может затормозить мгновенно. Это тем более верно для космического корабля, на который не действуют силы трения, но... мы отвлеклись.

Реализовать такое поведение в игре достаточно просто. В момент нажатия клавиши в функции изменения координат необходимо увеличить и значение велосити, а при отпускании клавиши – уменьшить его. Например, так:

```
int dir=1;
if (LEFT){velocity=4.5; dir=1; x-=3;}
if (RIGHT){velocity=4.5; dir=-1; x+=3;}
if (velocity>0) {
  velocity-=0.1;
} else {
  velocity=0;dir=0;
}
x=x-velocity*dir;
```

Переменная dir просто задает направление движения: **+1** – налево, **-1** – направо.

Обратите внимание на последнюю строчку – именно она обеспечивает «тормозной путь» после отпускания клавиши, а заодно и меняет знак приращения в зависимости от значения dir. Физическая модель заключена в блоке if () {...} else {}: если мы не поддерживаем нужную скорость, удерживая клавишу нажатой, она (за счет действия сил трения, надо думать) постепенно уменьшается. Поскольку трение в космосе не слишком велико, коэффициент затухания можно выбрать малым – подберите его опытным путем, чтобы играть было интересно.

Оживляем персонажей

Анимация – один из ключевых аспектов, на который нужно обратить самое пристальное внимание. Конечно, можно создать игру, где будут только статические персонажи – но ведь это же несерьезно!

Эффект анимации, как известно, создается последовательным отображением незначительно отличающихся друг от друга кадров. Таким образом мы можем украсить игру роскошными взрывами. Если вы внимательно читали предыдущие номера **LXF**, то надеюсь, не пропустили руководство по работе с 3D-редактором *Blender* (см. LXF87/88-LXF91 на DVD) – здесь он окажет вам неоценимую помощь.

Ниже представлен пример, демонстрирующий основы анимации. В данном случае ролик формируется из четырех кадров:

```
#include "ingame.h"
int main(int n, char **s) {
  screen(640,480);
  loadsprite (10,"01.png");
  loadsprite (11,"02.png");
  loadsprite (12,"03.png");
  loadsprite (13,"04.png");
  int i=10;
  while (GAME){
    sprite(i, 200 , 200);
    i++;
    if (i==14){i=10;}
    fx();
```

```
  }
  return 0;
}
```

Кадры занимают слоты с 10 по 13. Эффект достигается циклической сменой номера текущего спрайта, хранящегося в переменной i. Но попробуйте запустить пример – и вы увидите одну неприятную деталь: кадры сменяют друг друга слишком быстро. Избавиться от нее можно одним способом – введя искусственную задержку. В простейшем случае это достигается так:

```
int anim=10;
int c=0;
while (GAME){
  c++;
  if (c==20){anim++; c=0;}
  if (anim==14){anim=10;}
  sprite(anim, 200 , 200);
  fx();
}
```

Теперь номер текущего спрайта хранится в anim, а c – это счетчик для замедления. Кадр не меняется до тех пор, пока c не достигнет определенного значения – в нашем примере, **20**. Изменяя данный порог, можно управлять скоростью анимации, но она будет аппаратно-зависимой: при переходе на более мощный компьютер «человечки на экране начнут смешно махать руками» (если вы когда-либо играли в Диггера на 80386DX-2, вы меня поймете). Более правильным способом будет привязать задержку к абсолютному интервалу времени – скажем, менять кадр каждую 1/24 секунды. Этого можно добиться с помощью функции SDL_GetTicks(), возвращающей число миллисекунд, прошедших с момента инициализации библиотеки, и я оставляю данный вопрос вам на самостоятельное изучение.

Трели летнего утра

До сих пор наш игровой процесс протекал на равномерно закрашенном черном фоне. Для космической стрелялки это, может, и не плохо, но для большинства других игр не подходит. Поэтому в файле **ingame.h** определена функция colorfon(), принимающая три параметра: значения красной (R), зеленой (G) и синей (B) составляющей цвета фона. Подсмотреть значения RGB для интересующего вас цвета можно в палитре любого графического редактора, будь то *KPaint* или *GIMP*.

Если в качестве фонового рисунка разместить полупрозрачный спрайт, совпадающий по размерам с окном игры, то, динамически меняя цвет фона, можно добиться интересных эффектов: вечерней зари или рассвета. Главное здесь – ваша фантазия. Видели голубую пыль в космическом облаке в *StarFighter* на **LXFDVD**? Она получается именно так.

Не меньшее значение, чем красивый фон, имеет и звуковое оформление. Для этих целей предусмотрена функция loadsound(имя_файла, номер_слота), загружающая любой звук в формате wav в один из 500 доступных слотов. Воспроизвести звук можно функцией sound(), принимающей единственный аргумент – номер слота. Давайте добавим к игре рев идущих на форсаж моторов и грохот выстрелов (кстати, а вы знали, что в безвоздушном пространстве звук не передается? Да? Ну тогда считайте все это «литературным приемом»).

```
#include "ingame.h"

int main(int n, char **s)
{
  screen(1024, 768);
  loadsound("boom.wav",15)

  while (GAME) {
    if (LEFT) {sound(15);}
    fx();
  }
  return 0;
}
```

Подобным образом вы можете озвучить все действия в игре. А как же обстоит дело с фоновой музыкой? И на этом фронте у нас все в порядке. Функция loadmusic(название_файла, номер_слота) загружает мелодию в память. Поддерживаются форматы **mid**, **mod**, **xm**, **it**, **s3m**, **wav** и другие. Функция music(номер_слота) воспроизводит загруженную ранее мелодию, но имейте в виду – вызывать ее необходимо вне главного цикла, примерно так:

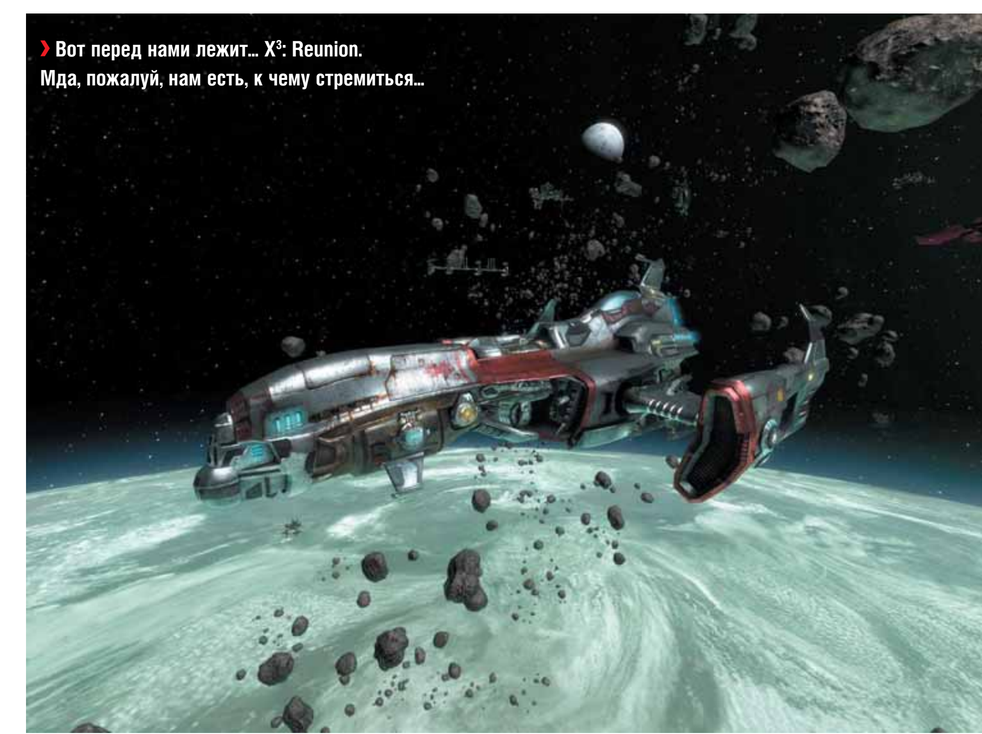
```
#include "ingame.h"
int main(int n, char **s)
{
  screen(1024, 768);
  loadmusic("level1.mod",1)
  music(1);
  while (GAME) {
    fx();
  }
  return 0;
}
```

Все вместе

Вот мы и подошли к тому моменту, когда осталось только одно – сесть и написать игру. Никакие отговорки, как вы понимаете, теперь не помогут. В ваших силах создать программу с отличным звуком и графикой. Файл **ingame.h**, который мы использовали во всех наших примерах, был написана автором специально для статей этой серии и распространяется на условиях GPLv2. Все входящие в него функции (LXF98) абсолютно прозрачны и доступны для редактирования и изменения.

В **ingame.h** не предусмотрена обработка ошибок. Это сделано по следующим причинам: во-первых, дополнительные проверки сделали бы код более громоздким и менее понятным, что для обучающего материала неприемлемо. На мой взгляд, проще потом внести пару собственных строчек. Во-вторых, игровая программа не сможет быть запущена, если какие-либо файлы, входящие в комплект, повреждены или отсутствуют. Но, положив руку на сердце, скажите, действительно ли вы хотите, чтобы кто-то запустил вашу игру с поврежденной или недостающей графикой?

Я также могу предположить, что у особенно заинтересовавшегося читателя возникнет желание разобрать файл **ingame.h** на кусочки, дабы понять, как же все устроено. Тогда вам придется углубиться в таинства библиотеки SDL – а это настолько же большая тема, насколько коротка наша серия. И сейчас, думаю, пришло время подумать о создании игр и других вещах: о башмаках и сургуче, капусте, королях, и почему, как суп в котле, кипит вода в морях. Да, не забывайте присылать ссылки на сделанные вами игры на letters@linuxformat.ru – возможно, мы даже разместим наиболее удачные экземпляры на одном из **LXFDVD**! LXF



› Вот перед нами лежит… X[3]: Reunion. Мда, пожалуй, нам есть, к чему стремиться…



Стратегия внедрения СПО в учреждениях образования



Формирующаяся в учреждениях образования инфраструктура информационных технологий базируется на широком внедрении в процессы обучения, управления и коммуникации современных технических средств обучения и обучающих ресурсов. Создание комплексной информационной среды обусловлено реализацией приоритетного национального проекта «Образование» по направлению «Внедрение современных образовательных технологий». В рамках реализации мероприятий «Развитие технической основы современных информационных образовательных технологий» приобретает актуальность проект «Обеспечение доступа к сети Интернет образовательным учреждениям (ОУ) Российской Федерации».

По состоянию на 28 мая 2007 года широкополосный доступ к сети Интернет получили 35 125 общеобразовательных учреждений Российской Федерации. До конца 2007 года будут подключены оставшиеся 17 627 общеобразовательных учреждений, большинство из которых – сельские школы. Планируется, что за два года (2006-2007) все российские школы получат неограниченный широкополосный доступ к Всемирной паутине, что позволит выровнять образовательные возможности всех российских школьников и повысить качество образовательных услуг.

В 2008–2009 годах все российские школы будут обеспечены бесплатным трафиком за счет средств федерального бюджета, об этом сообщает пресс-служба Министерства образования и науки РФ. В настоящее время рассматривается возможность подключения к Интернету всех учреждений начального и среднего профессионального образования, а также учреждений дополнительного образования.

В результате реализации проекта будет сформирована необходимая техническая основа для внедрения современных образовательных технологий обучения. Министерством образования и науки РФ разработан комплекс мер по повышению эффективности использования информационно-коммуникационных технологий в учебном процессе. Ведется разработка электронных образовательных ресурсов нового поколения. Они призваны обеспечить реализацию образовательных программ по таким общеобразовательным предметам, как физика, химия, биология, иностранный язык, география, мировая художественная культура и искусство, история, русский язык, естествознание, математика. Все создаваемые образовательные ресурсы ориентированы на работу через сеть Интернет.

При формировании системной основы использования информационных технологий и продуктов необходимо принимать во внимание их специфику. Одним из парадоксов информационных технологий можно назвать негативное влияние высокого темпа их развития. Высокая динамика развития ИТ приводит к быстрому моральному устареванию техники. Для образовательных учреждений это означает необходимость учета в схеме затрат средств на регулярную модернизацию компьютеров и программного обеспечения.

Так, по данным Счетной палаты РФ (опубликованы в журнале «Национальные проекты» и относятся к промежуточным итогам реализации приоритетного национального проекта «Образование»), с наибольшими трудностями столкнулись начальные и коррекционные школы: компьютерная база в них устарела и недостаточна для подключения к Интернету.

Информационная и коммуникационная прозрачность определяет большой пул уязвимостей для системы образования – объемы сетевых ресурсов образовательного направления настолько велики, что уровень достоверности публикуемой информации достаточно трудно отследить.

Основным недостатком, по мнению аудитора Счетной палаты РФ Александра Назарова, явилось отставание в обеспечении национального проекта нормативными правовыми документами.

Стратегическая конкурентоспособность в условиях глобализации зависит от оптимизации и модернизации индустриального сектора и, одновременно с этим, от создания информационно-технологического уклада. Переход мирового сообщества к качественно новому технологическому укладу, базирующемуся на сетевом развитии производства и управления, на массовом использовании информационно-коммуникационных технологий, определяет новую ступень современной цивилизации.

Применительно к системам образования взаимопроникновение процессов информатизации и создания системы менеджмента качества проходит по следующим направлениям:

- программно-техническое обеспечение системы менеджмента качества образования;
- информационное сопровождение контрольно-инспекционной и надзорной деятельности в сфере образования;

- информационное сопровождение лицензирования и государственной аккредитации образовательных учреждений;
- повышение эффективности управления системой образования за счет внедрения ИКТ в практику работы органов управления образованием и образовательных учреждений;
- развитие интегрированных информационных систем (информационное обеспечение системы менеджмента качества образования):
- создание информационного пространства общероссийской системы оценки качества образования;
- развитие телекоммуникаций в интересах управления образованием;
- межуровневая, межрегиональная, межотраслевая и международная интеграция информационных систем;
- развитие информационных систем единого государственного экзамена;
- внедрение в процесс информатизации системы менеджмента качества;
- стандартизация информационных технологий в сфере образования;
- нормативно-правовое регулирование информатизации образования;
- обеспечение информационной безопасности.

Формирование отечественной системы образования на основе интеграции широкого спектра информационных и технологий ставит перед руководителями разных уровней системные вопросы. Процесс консолидации региональных компонентов в единое информационное общество создает предпосылки для систематизации, популяризации и разъяснения сведений правового, экономического, технического и методического характера по данной проблематике.

Предметом выходящей в эти дни книги Г.Ю. Пожариной и А.М. Поносова «Стратегия внедрения свободного программного обеспечения в учреждениях образования» является использование свободного программного обеспечения в качестве базы информатизации образовательного учреждения.

Книга ориентирована на руководителей образовательных учреждений общего и полного среднего, начального и среднего профессионального, высшего профессионального образования, на руководителей подразделений информатизации образовательных учреждений.

Раскрывая концепцию миграции образовательного учреждения на свободное программное обеспечение, авторы обращают внимание на правовые аспекты использования программного обеспечения, экономическую целесообразность использования свободных программных решений, обосновывает значение процесса в формировании системы менеджмента качества учреждений образования.

В виде конкретных программных решений описаны компоненты системы информатизации образовательного учреждения на основе свободного программного обеспечения.



Издание снабжено справочным материалом нормативно-правового, технического и методического характера.

Приложения на двух компакт-дисках включают:

- сборник свободного программного обеспечения для применения в учебном процессе, акцент в котором сделан на кроссплатформенном программном обеспечении;
- дистрибутив Mandriva One, предназначенный для ознакомления с работой операционной системы Linux без предустановки на жесткий диск компьютера,
- сборник актуальных нормативных документов в области информатизации образования, которые составляют полную информационно-правовую базу для работы подразделений информатизации учреждений образования.

«У самовара» с Александром Поносовым



Поводом для написания этой статьи стали разговоры, переписка с директорами школ, учителями о предстоящем внедрении пакета свободных программ. Поневоле у меня сложился собирательный образ человека, который как-то занимается (должен заниматься) информатизацией учебного заведения. Этакий «Фома неверующий» от образования. Его аргументы против перехода на GNU/Linux уложились в семнадцать пунктов, но список явно неполный. Комментируют «подводные камни», помогают в борьбе с «нездоровым скептицизмом и консерватизмом» мои «френды» и читатели LJ, сам я высказываюсь под ником *alex_ponosov*.

Три года счастья или Семнадцать сомнений Фомы

1 Все это не более чем политическая игра. Пошумим, деньги на эксперимент освоим, потом опять на поклон к корпорациям. А Linux останется конструктором для любителей.

last-skif: А у нас разве бывает иначе? Половина выделенных денег сразу же уйдет на «особые комиссии» которые будут выдавать «сертификаты» для особо приближенных фирм, которые получат эксклюзивное право на установку и последующее сопровождение школьных ПК. Никакого права выбора у школы «где покупать/обслуживаться» не будет. И никого не будет волновать, что точно такой же ПК в соседнем магазине стоит на 50 % дешевле «навязанного».

Pzz: Возможно. Однако если так рассуждать, вообще ничего делать не надо. Вообще. Каждый должен следовать в первую очередь своему долгу и убеждениям, даже если все вокруг – политическая игра.

alex_ponosov: Процитирую классика: «Политика – это сконцентрированная экономика». Не забывайте, что кроме школ, есть множество бюджетных организаций, на которые не распространяются академические лицензии. Трудно сказать, в какую сумму обойдется государству наведение лицензионной чистоты в своих пределах, если отказаться от внедрения свободного ПО. Бизнесмены тоже деньги считать умеют. Так что, если это игра, то игра с расчетом, и вряд ли она останется незавершенной.

2 На высоком уровне заявлено, что на все школьные компьютеры установят единый пакет програм-



много обеспечения, лицензии на которое будут приобретены на три года у Microsoft и других международных производителей на льготных условиях. Зачем в таком случае время и силы тратить?

Test-life: Сначала нам продадут за 3 копейки «затравку» в школы. А потом свои, родные, специалисты будут зарабатывать на русификации, установке и обслуживании «забугорных» программ, получая 1–10 % от дохода с ПО, а остальные денежки будут утекать за границу.

Примерно также наркоманов «подсаживают»: 2–3 «подарочные» дозы, а потом где хочешь, там и бери деньги.

alex_ponosov: Вообще-то, это счастье на три года. А потом учебные заведения должны сделать выбор: на свои деньги закупать лицензии или пользоваться свободными программами. Готовь сани летом....

3 У нас в школе все программы лицензионные, все прекрасно работает, так что ничего нам более не надо.

Pzz: Возможно, школам, у которых и так все в порядке, стоит подождать. Но все равно скоро переходить на Windows Vista, так почему бы не на Linux?

alex_ponosov: Те программы, на которые у вас имеются лицензии, лет через пять безнадежно устареют. А ведь школа должна идти в ногу с техническим процессом, чему способствует использование СПО.

К тому же, при использовании проприетарного ПО никогда нельзя быть до конца уверенным, что вы не нарушили чьи-то авторские права.

А как быть с учениками? Обучаясь в школе на лицензионной Windows, на домашней машине они поставит эту же ОС..., потому что у нас, например, 98 % семей учащихся относятся к категории малообеспеченных. Надеюсь, мысль понятна? И что после этого скажет совесть?

4 Снаряд дважды в одну воронку не падает. После такого шумного дела прокуратура вряд ли снова пойдет в школы с проверками.

Pzz: Это не аргумент. Что, после первого случая школы могут наплевать на лицензии?

alex_ponosov: Скорее всего наоборот, школы будут проверены и не раз, потому что поставка программ осуществляется в рамках национального проекта.

5 Зачем учить детей тому, что им вряд ли пригодится в жизни? Продукты Microsoft – стандарт де-факто. Значит, выпускник должен уметь ими пользоваться.

Test-life: Отлично. А сегодня дети, мы познакомимся с Windows. У нее почти все также. Отличия тут, тут и тут. Вот вам тестовое задание. Кто хочет узнать побольше о Windows, Apple (не хлебом единым...) – записываемся на факультатив.

Muromec.org.ua: Затем, что школа – не ПТУ. На уроках труда и физкультуры тоже учат тому, что ребенку не пригодится. Ну и скажу известную банальность: если правильно учить, оному выпускнику будет все равно, с чем работать.

alex_ponosov: Идеал – мультиплатформенный школьник, но не за счет натаскивания, а за счет хорошего знания теории. А обучать на Linux. Почему? А почему в автошколах обучение вождению не проводят на машинах с автоматической коробкой передач?

6 В дистрибутивах Linux мало драйверов. Практически все «железо» ориентировано на Windows.

Pzz: Практически все популярное оборудование отлично работает с Linux. Проблемы имеются у слишком новых ноутбуков (но у них проблемы и с Windows бывают) и у всякой экзотики, типа 10-гигабитных сетевых контроллеров. Но вряд ли что-то из этого встречается в школах.

Muromec.org.ua: Спрос рождает предложение. в условиях массового перехода школ целой страны на Linux производители вышеперечисленных вещей быстренько подсуетятся – им деньги зарабатывать нужно.

Анонимно: Полуправда. Действительно, меньше. С некоторым оборудованием (основное: win-модемы, часть принтеров, некоторые ноутбуки – действительно, нет) Но Вы правильно ставите вопрос о целесообразности. Я думаю: раз уж мы взялись продвигать Linux в школы России, то необходима будет и какая-то сертификация на совместимость со школьным дистрибутивом. И списки протестированного оборудования.

alex_ponosov: Некоторые проблемы с оборудованием появились: Winmodem, МФУ, еще не пробовал устанавливать Linux на старенькие машины. Думаю, это все решаемо при хорошо организованной технической поддержке разработчиков. А в дальнейшем нужно серьезно отнестись к закупкам оборудования, обязательно его тестировать. Например, когда я заказывал компьютер домой, то сразу предупредил сборщиков об ОС, результат: ни одной заморочки, потому что проверили на трех дистрибутивах.

А если в школе есть что-то особое, нетипичное для других, возможно придется по-прежнему использовать Windows, а в это время запустить головоломку Linux-сообществу.

7 За предыдущие годы скопилось множество документов *Excel*, *Word*. Как мы будем работать в *OpenOffice*, если все отчеты, прочая переписка с государственными структурами ведется в этих форматах?

Last-skif: Отлично будем работать. Уже почти год как у меня стоит *OpenOffice*, ни одной проблемы с открытием как у меня, так и моих файлов у других мне не встретилось. А большая половина таких «параноиков» и вовсе не способна с первого раза определить по виду, что установлено на ПК: *MSO* или *OOo*.

Pzz: *OpenOffice* отлично открывает документы, созданные старыми версиями *Word* и *Excel*. Мы ведь о них говорим, не так ли?

Анонимно: Не решена проблема, увы. Есть масса ПО и документов, которые без *MS Office* не работают. И самым разумным решением зачастую является подход: «Собрать весь зоопарк на 1–2 компьютерах с Windows»

alex_ponosov: При установке *OpenOffice* в школе единственное, что мне пришлось сделать – это объяснить сохранение документов в формате DOC. Сам почти год пользуюсь *OpenOffice*, с проблемой совместимости столкнулся один раз, на прошлой неделе получил вложение application/x-msword (071220.doc), открылась абракадабра, попросил повторить в другом формате.

И вообще, если на государственном уровне будет принят стандарт ODF, проблемы будут не у пользователей Linux.

8 Все цифровые образовательные ресурсы работают только под Windows.

Анонимно: К сожалению, это правда. Но правда и то, что образовательных ресурсов и под Windows мало. Будем нарабатывать, будет интерес – появятся и ресурсы.

Last-skif: Образовательные курсы... Я не ручаюсь за всех, но большинство попадавшихся мне отлично работало в среде эмуляторов или вовсе представляло собой набор экранных снимков с описанием в HTML. А уж оно точно 100% совместимо. В общем, речь об этом уже была, я свое мнение высказал: надо учить принципам работы, а не тому, где находится нужная кнопочка в конкретном интерфейсе.

alex_ponosov: По большому счету, большинству пользователей без разницы, какая ОС стоит на компьютере. Главное, чтобы работали прикладные программы. В школе – это электронные образовательные ресурсы (ЭОР). Я перепробовал кучу дисков с обучающими программами под WINE 0.9.44, не работают четыре из пяти. Плохо. Справедливости ради надо сказать, что некоторые из них некорректно работают и под Windows, а несколько штук вообще надо выкинуть, устарели.

Казалось бы, в свете последних событий все разработчики должны осознать, что ЭОР нового поколения должны быть кроссплатформенными. Ан нет. Заглянул на сайт Федерального центра информационно-образовательных ресурсов (ФЦИОР, адрес fcior.edu.ru, и второй адрес – eor.edu.ru) Хотел посмотреть учебный модуль по истории, так ОМС – открытая образовательная медиасистема – не захотела признать Linux в качестве клиента. 10 тысяч модулей... Кто и как их запустит под Linux, вопрос остается открытым.

9 А как быть с бухгалтерскими программами? Вот уж где недопустимы риски!

Kaf-nevod: Один из крупнейших разработчиков бухгалтерских программ – фирма 1С с версии 8.0 начала поставлять 1С-сервер как для Windows, так и для Linux.

Мы только, что закончили установку 1С: Предприятие под Linux для довольно крупного предприятия.

Сервер HP Proliant, два процессора Xeon, 6 ГБ оперативной памяти. Сервер обеспечивает работу до 50 пользователей. Кроме поддержки сетевых 1С-клиентов, сервер работает в терминальном режиме, обеспечивая запуск Windows-клиентов 1С в Linux-среде. Использование NX-технологии позволяет работать в графическом терминальном режиме при скорости обычного dial-up модема – 5–6 Kb в секунду. По крайней мере, мы большую часть работы делали в этом режиме через Интернет, находясь в 200 километрах от заказчика. Причем Windows-клиент работает в KDE-среде, что потом облегчит перевод клиентов полностью на Linux. Мы просто заменяем Windows на Linux или ставим бездисковый терминал...

Анонимно: К учебному процессу отношения не имеет. Если речь о школьной бухгалтерии – никто не мешает на 1–2 компьютера и Windows купить. Вполне



достаточно XP Home OEM – при покупке вместе с компьютером.

alex_ponosov: На первых порах самое разумное – оставить бухгалтерию в покое, т.е. на Windows. Будет обеспечена надежная работа бухгалтерских программ под Linux, тогда и решение принимать. Если дойдет дело до включения бухгалтерии в школьную сеть (например, для работы с «Хронографом»), то поводов для беспокойства тоже нет: Windows и Linux, по уверениям специалистов, прекрасно уживаются в одной сети.

10 Все говорят о низкой стоимости СПО. А сколько денег уйдет на техническую поддержку, переобучение?

Pzz: А на поддержку Windows сколько денег уходит? И кому, собственно, они достаются? Дипломированным специалистам? Не верю. Скорее всего, Васе из подворотни. Ну так найдете Петю, со знанием Linux.

Test-life: Я бы ответил встречным вопросом: «А сколько уйдет на обучение и поддержку в случае с MS-продуктами?». Поинтересуйтесь, сколько стоит час работы сертифицированных специалистов под MS.

alex_ponosov: Проблемы технической поддержки и уровня квалификации учителей информатики назрели давно, и решать их пришлось бы и без Linux. А так, просто появился шанс заняться информатизацией школ комплексно, укрепить слабые места. Отправить учителя информатики на курсы Linux, сначала для пользователей, а потом для системных администраторов – решение весьма полезное, даже если ваше заведение категорически против перехода на Linux.

11 Linux — очень сложная система, с которой может разобраться только профессионал. Для детей это будет слишком тяжело. Им придется изучать много лишних вещей.

Test-life: Вы хотите сказать, что с Windows вы разобрались? Люди ходят на сильно платные курсы, сдают многоступенчатые экзамены, чтобы получить звания по знанию Windows. А подавляющее большинство знает «минимум миниморум», да и то, некоторые моменты в голове как молитвы: произнес – получилось.

В учебном плане стоит обучение детей установке ОС? Настройке ПО?

Насколько я понимаю, дети хранят файлы и работают в русскоязычных прикладных программах. Эти действия не сильно различаются в разных ОС. И также, ОС не является залогом стройности или убогости мысли программистов, пишущих прикладной софт.

Так что, будет тяжело детям в Linux? Какие лишние вещи им надо будет изучить при работе с настроенной системой?

alex_ponosov: Освоить Linux «снаружи» несложно, могу судить об этом по своим детям, они это сделали самостоятельно. Сдается мне, мы недооцениваем наших учеников – вещи, которые кажутся взрослым лишними и сложными, они схватывают на лету. В этом отношении при использовании СПО у желающих появляется возможность изучить программы изнутри, а значит увеличиваются шансы стать профессионалами.

12 Бесплатные программы, которые пишутся неизвестно кем, не могут быть надежны, а Windows надежна, потому что мы за нее заплатили. В случае проблем с Windows мы всегда можем предъявить претензии Microsoft.

Muromec.org.ua: На разработку «Школьного дистрибутива» государство выделяет 70 млн., кроме того, разработку критически важных частей GNU/Linux оплачивают зарубежные корпорации.

Drcatspb: Среди этих «неизвестно кого» присутствуют такие гиганты, как HP, IBM, Novell.

alex_ponosov: «Школа предъявляет претензии к корпорации XXX за срыв учебного процесса...» Ненаучная фантастика.

13 А как разобраться, какой дистрибутив Linux ставить? Их столько, что глаза разбегаются. Вот будет один, «вылизанный» как Windows, тогда подумаем.

Анонимно: Не нужно выбирать на уровне школы. Нужен один стандартный дистрибутив для школ. Какой? Не столь важно – лишь бы у всех был одинаковый.

Анонимно: Проблемы решаемы. Стандартизацией дистрибутива (и оборудования – вообще идеальный вариант), организацией нормальной техподдержки для школ и написанием, выпуском методичек. Цель: не оставлять каждую школу со зверем по имени «Linux» наедине, а объединить усилия.

alex_ponosov: Один дистрибутив – это хорошо на первых порах. По мере освоения захочется некоего разноообразия. Это нормально. Никто из историков, к примеру, не падает в обморок от обилия учебников, а выбирает инструмент для решения учебной задачи. Так и с дистрибутивами. Пусть на коробках стоят грифы «Допущен», «Рекомендован», а дальше – на усмотрение школы.

14 У дистрибутивов Linux очень сложная процедура установки. Надо читать разные руководства, потом, как говорится, «поработать напильником»

Pzz: Это неправда. Современные дистрибутивы Linux ставятся очень легко.

Анонимно: С рабочей станцией справляется ученик 5-го класса. С сервером удалось разобраться за несколько вечеров. «Точить» пришлось сугубо в штатном режиме.

last-skif: Для специалиста из фирмы – разница нулевая. А детей никто ставить Linux на уроках не принуждает. А уж как приходится дорабатывать напильником Windows (патчи к которой занимают больше ее дистрибутива), можно рассказывать бесконечно.

alex_ponosov: Шаманские пляски, похоже, уже надо включать в многотомник «Мифы и легенды Linux».

Здесь не надо много слов – дайте сомневающемуся диск, и пусть попробует, убедится сам.

15 А кто нам поможет? Во всей округе специалиста по Linux днем с огнем не сыщешь.

Last-skif: Слухи о сверхстоимости Linux-спецов преувеличены. Я бы и сам отлично пошел на стандартную оплату. Вот только пора это никому не нужно. Нет спроса. А переучиться – дело пары недель. Общие принципы везде одинаковы.

Более того, Linux-спецы намного более «молодежны» и намного менее консервативны, чем Windows-профи. В конце-концов, я лично учился информатике у людей, которые были просто переучены с «математики» и прочих наук. И ничего – дураком себя совсем не считаю.

Будет спрос – будет и предложение. Если что – вырастите Бабу Ягу в своем коллективе.

alex_ponosov: Большинство школ сейчас как никогда нуждаются в качественном ИТ-сервисе: школьная сеть с общим доступом в Интернет, автоматизация управления и другие звенья информатизации требуют помощи квалифицированных специалистов. Скажу точно: таковые имеются. Осталось заложить в бюджет школы необходимые средства. А дальше самые разные варианты: свой инженер, удаленное администрирование, приходящий системный администратор и т.п.

16 В школе программы, учебники, методички под Windows. Школа просто разорится закупать другие.

Анонимно: Было бы чего закупать! Пока, кроме учебно-методического комплекта Угриновича, для школ ничего нет. А у кого он есть, и закупать больше ничего не надо.

alex_ponosov: Конечно, идеальный вариант – предусмотреть централизованное обеспечение школ учебно-методическими комплектами в рамках национального проекта. Если рассчитывать на свои силы, то действительно, можно заниматься по имеющимся учебникам, проблемы только с практическими занятиями, некоторое время придется терпеть неудобства. И закупать новые, деньги выделяются, обновлять фонд учебников все равно когда-то надо.

17 Календарно-тематическое планирование, поурочные разработки - это что же все переделывать? Ни за что!

Pzz: Возможно. С другой стороны, они все равно устаревают.

Test-life: Стратегические интересы иногда идут вразрез с локальными.

Last-skif: Что же там такого сложного? Find&Replace слов Windows на Linux? Или, процесс обучения поиском в Windows настолько сильно отличается от Linux? Или, текстовые редакторы там с марсианским интерфейсом?

Анонимно: Если вы спрашиваете «чего там сложного», то вы просто некомпетентны. Чем качественнее уроки – тем сложнее их перестраивать.

Все в самом деле плохо. Автозамена «не проходит» во многих случаях. У меня есть один такой пример – с VBA. Причем, что самое печальное, переход на Linux бьет наотмашь по лучшим учителям. По тем, кто пытался делать что-то свое – за рамками школьной программы.

alex_ponosov: А вот это уже начало школьного «холивара». «Чему учить, как учить?» – «ломать копья» по поводу преподавания информатики будем в следующей статье. LXF





Нижегородский технический

Сейчас многие переходят на GNU/Linux – жизнь заставляет. А технически грамотные люди сделали это уже давно. Они не называют это переходом, потому что просто используют GNU/Linux там, где им это нужно. LXF беседует с одним из таких людей – **Дмитрием Бондиным**.

Евгений М. Балдин (LXF): **Дмитрий Валерьевич, расскажите, пожалуйста, о себе.**

Дмитрий В. Бондин (ДБ): В 2001 году я закончил Нижегородский государственный технический университет. Учился на факультете информационных систем и технологий (ныне Институт радиоэлектроники и информационных технологий). Специализировался на кафедре прикладной математики (ПМ) по специальности «прикладная математика».Сейчас выступаю в роли ведущего программиста кафедры ПМ НГТУ (веду курсы «Структуры данных и алгоритмы», «Базы данных», «Распределенные вычислительные системы»), аспиранта кафедры ПМ НГТУ по специальности «математическое моделирование, численные методы и комплексы программ», инженера-программиста ООО «Теком» и приглашенного преподавателя учебного центра «IT-Эксперт», курс «Основы администрирования Linux».

LXF: **Кого готовит Ваша кафедра?**

ДБ: У нас два основных направления: математическое моделирование (в основном, гидродинамика и экономика) и научная подготовка студентов – далее аспирантура, ученые степени и т.д.; и информационные технологии, то есть инженерное направление – далее работа в ведущих IT-компаниях Нижнего Новгорода.

Естественно, возможны комбинации (работа в IT-компании и научная работа одновременно). У кого на что сил хватает.

LXF: **Что в используется в процессе обучения?**

ДБ: Аппаратная база кафедры:

- 2 лабораторных зала по 15 ПК.
- 1 старый лабораторный зал с десятью SparcStation 10/20 – некуда ставить (улыбается).
- «Малый» экспериментальный кластер (10 ПК, Pentium 4, Gigabit LAN).
- «Большой» рабочий кластер (20 ПК, Pentium 4, Gigabit LAN) – не очень, конечно, большой, но больше «малого» (улыбается).
- Несколько различных серверов (PC, Sun Sparc Station 10/20).
- Рабочие компьютеры инженеров и преподавателей.

Все это хозяйство, естественно, находится в кафедральной сети.

В настоящий момент основной платформой при обучении студентов является Fedora Core 4. Естественно, при необходимости используется и Windows. Принцип выбора платформы: если можно, не перепрыгивая через голову и не теряя принципиальных моментов, сделать курс под GNU/Linux, то его нужно сделать под GNU/Linux. Если нет, то покупается необходимое ПО для Windows.

Программные пакеты используются самые разнообразные, так как курсов много, и в каждом из них –

свой набор используемого ПО. Например, практически во всех курсах применяются *GCC*, *GNU Make*, *automake/autoconf* сотоварищи. В качестве СУБД используем *PostgreSQL* и unixODBC, для обучения численным методам – *GNU Octave*, *SCILAB*, *gnuplot*, для распределенных вычислений – *MICO* (http://www.mico.org/) .

Из своих наработок можно упомянуть MPI-кластер, который активно используется командой под руководством доктора физ.-мат. наук А. А. Куркина для обсчета различных задач гидродинамики.

LXF: По каким причинам возник вопрос перехода на GNU/Linux?

ДБ: О, давно это было. Я тогда учился на третьем курсе, то есть это, получается, 1998 год. Пришли учиться после лета, в вместо DOS и Borland C 3.1 стоит RedHat 4.2 и GCC.

Затеяли это «безобразие» молодые (тогда) инженеры кафедры Бабкин Эдуард Александрович и Зубов Михаил Львович. В настоящий момент они преподают в Нижегородском филиале Высшей школы экономики.

Основной целью перехода была стабильная рабочая лаборатория, так как DOS'овский зал приходилось переставлять чуть-ли не каждую неделю. Студенты ведь люди увлекающиеся, и им постоянно хочется везде поковыряться. Соответственно, зал был переведен на многопользовательскую ОС с четким разделением прав. Программное обеспечение на тот момент в общем-то устраивало: GCC как основное средство разработки (C/C++/Fortran77), Tcl/Tk (GUI для программ), gnuplot (визуализация расчетов), *Postgres95*, потом *PostgreSQL* (основная СУБД) и так далее. Несколько машин с неисправными жесткими дисками грузились прямо с сервера! Сначала надо было загрузочные дискетки вставлять, потом BootRom прошили в сетевые карты и стало совсем хорошо. Сеть тогда была еще была на базе коаксиального кабеля. Машины были уровня 386DX-486DX, 2-4 МБ ОЗУ. Потом новый зал купили с Pentium 166, 8 МБ ОЗУ, 1 ГБ жесткими дисками, Ethernet 10M, сеть сделали гибридную: от «гагарина» до зала – коаксиал, а по залу – витая пара звездой. «Гагарин» – это сервер у нас так когда назывался; главный кафедральный файловый сервер и сейчас так называется – традиция (*улыбается*).

LXF: Какой вы выбрали дистрибутив GNU/Linux и почему?

ДБ: Насколько я помню, было как-то так: Red Hat 4.2, 5.0, 5.2, 6.2, 7.2, 8, 9, FC2, FC4. Сначала выбрали Red Hat, потому что он оказался под рукой. Потом сила привычки, как-то все «красная шапочка» да «красная шапочка». Сейчас на лето планируется переход с FC4 на что-то более свежее, и боюсь, что Fedora уже не будем ставить. Начиная с FC2, все крайне сырое, приходится море патчей накладывать, и уже порядком поднадоело. Пока не знаем, что будет выбрано: этим занимаются три человека (включая и меня), имеющих, соответственно, три разных мнения: Slackware, Gentoo или самосбранная система на базе LFS. Посмотрим, что получится... Вот сказал про LFS, и сразу мысль – это ж сколько студентов можно делом занять без отрыва от учебного процесса!

LXF: Как на GNU/Linux реагируют студенты?

ДБ: И девять лет назад, и сейчас, ситуация примерно одинаковая: половина студентов дергает мышкой и горько плачет по привычным окнам, вторая половина изучает систему глубже, понимают всю прелесть и остаются в стане GNU/Linux. Из тех, что из второй половины, получаются либо отличные инженеры, которых с руками отрывают в IT-компаниях, либо грамотные научные работники. А из первой – как повезет.

LXF: Возникают ли проблемы социального плана у преподавателей/администрации? Как они решаются?

ДБ: Проблемы, конечно же, были и есть. Особенно у преподавателей, которые, как бы это помягче сказать, прикипели к неким программным пакетам, причем не первой свежести, и ни сил, ни желания освоить что-то новое у них нет. В каждом конкретном случае – подход индивидуальный. На сегодняшний, практически все курсы читаются под GNU/Linux.



› «Большой» кластер и аспирант Андрей Лухнов.

LXF: Будучи студентом, вы на себе испытали переход на GNU/Linux. Образовательный процесс как-нибудь изменился?

ДБ: Нет, принципиальных отличий я не заметил.

LXF: Какие методические материалы сейчас используются в обучении?

ДБ: В рамках каждого курса имеются свои наработки. В основном, авторские.

LXF: А чего Вам сейчас не хватает для полного счастья?

ДБ: Свободного времени (*улыбается*).

LXF: Что бы Вы порекомендовали сообществу для того, чтобы процесс внедрения GNU/Linux в образование пошел активнее?

ДБ: Очень правильный принцип: не надо обучать студентов работе с программными продуктами, а надо изучать технологии, на/для которых эти продукты созданы. И сразу отпадает множество вопросов типа: «хочу уметь работать с MS SQL Server» и т. д. Научите человека реляционной алгебре, *SQL*, процедурным расширениям, интеграции с пользовательским ПО, и ему станет без разницы, что у него, *SQL Server* или *PostgreSQL*.

Соответственно, пожелать хочется осознания данного принципа максимально большим количеством людей, причастных к образовательной сфере. LXF



› Пока студент запускает «лабу»...



Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

1 Настроить загрузку с CD

В Мне ужасно понравилась ваша статья «Собери себе Ubuntu» в LXF97. Я выполнил все ее инструкции на моем ноутбуке HP DV9308nr, добавил прошивку BCM43xx в созданный ISO-образ, записал на диск, загрузился с него, и все заработало как ожидалось. Как мне отредактировать файл **menu.lst** в **/boot/grub/** в ISO-образе, чтобы передать ядру следующие параметры:

```
vga=791 pnpbios=off irqpoll nomsi nomce
```

Мой ноутбук без этих установок блокируется.

Филипп Шарбоно [Philippe Charboneaux]

О Ubuntu использует на своих CD не *Grub*, а *isolinux*. Его файл конфигурации – **isolinux/isolinux.cfg** на CD. Синтаксис файла будет вполне знаком тем, кто использовал *Lilo*. Вы можете скопировать содержимое CD в директорию **ubuntu-rebuild**. Теперь в файле **ubuntu-rebuild/isolinux/isolinux.cfg** найдите первый пункт меню, который в Ubuntu 7.10 выглядит так:

```
LABEL live
 menu label ^Start or install
 Ubuntu
 kernel /casper/vmlinuz
 append file=/cdrom/preseed/ubuntu.
seed boot=casper initrd=/casper/initrd.
gz quiet splash --
```

Строка **append** содержит параметры, передаваемые ядру при загрузке, как и такая же строка в **lilo.conf**, так что добавьте ваши опции в эту строку перед «--» на ее конце и сохраните файл. Теперь продолжите по учебнику, и ваш переделанный диск загрузится с нужными вам опциями. Еще есть проблемы? Дайте нам знать!



› Измените параметры загрузки вашего личного диска Ubuntu, отредактировав isolinux.cfg.

2 Блуждающий диск

В У меня USB-винчестер Seagate на 250 ГБ, подключенный к моему ноутбуку Dell Inspiron с Ubuntu 7.04. Распознается он нормально, но каждый раз при загрузке появляется в **/dev** в разных местах – где-то между **/dev/sdb** и **/dev/sdh**. Из-за этого автомонтирование разделов, которые я создал на диске, мягко говоря, усложняется.

Существует ли способ заставить этот диск всегда появляться как определенное устройство в **/dev**, чтобы я мог добавить его в **/etc/fstab** и не злобствовал, пытаясь отыскать, куда Ubuntu вздумалось засунуть диск сегодня, и затем вручную монтировал свои разделы?

И еще, есть ли способ добавить разделы в **/etc/fstab** так, чтобы, независимо от расположения их **/dev**, они монтировались в определенные точки в **/media**?



› USB – не самый скоростной путь передачи данных; даже USB 2.0 гораздо медленнее, чем Ethernet.

Наши эксперты

» Мы найдем эксперта на любой вопрос! вы получите ответ на все, от проблем с установкой или модемом до сетевого администрирования; главное – спросить!



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс
Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон
Пол – местный суперпрограммист. Он управится со всеми вашими проблемами по части web и баз данных.



Грэм Моррисон
Когда он не обозревает вершины программного обеспечения и не халтурит с MythTV, Грэм готов к ответам касательно любого оборудования и проблем виртуализации.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите нам по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru



Ubuntu монтирует разделы при старте, но большинство из них на 40 ГБ, и они получают не самые «говорящие» имена '40GB disc-1', '40GB disc-2' и т.д. Мне нужно, чтобы они автоматически монтировались в более осмысленные места.
Дэвид Шоу [David Shaw]

Можно велеть **fstab** монтировать разделы по их уникальным ID вместо традиционных узлов **/dev/sd**? Заглянув в **/dev/disk/by-id**, вы увидите символьные ссылки на различные разделы дисков с использованием их ID. Эти ID сохраняются неизменными, а ссылки на конкретные узлы обновляются каждый раз при старте *udev* (обработчик устройств в Linux). Вы могли бы использовать их в **/etc/fstab** вместо традиционных имен узлов, но вообще-то они предназначены не для использования со сменными дисками. Преимущество данного метода для постоянных дисков в том, что при добавлении разделов нумерация сбивается, но **/etc/fstab** будет продолжать действовать, и поэтому Ubuntu использует эту систему по умолчанию.

При использовании сменных дисков лучше вообще не прописывать их в **/etc/fstab**. В этом случае *Gnome Volume Manager* автоматически смонтирует их, как только обнаружит, при загрузке или в любое время позже. Но это как раз и приводит к вашей проблеме, и для обеспечения неизменности имен устройств мы воспользуемся другой функцией *udev*. *Udev* поддерживает правила, которые могут, среди прочего, задавать имена узлов **/dev** для специфических устройств или файловых систем, и система будет их помнить, так что вы сможете дать Вашим устройствам содержательные имена, типа **/dev/extdisk** для внешних дисков и **/dev/camera** для... ну, вы поняли идею. Для использования этих правил вам нужно обеспечить уникальность идентификации устройств; здесь поможет команда *udevinfo*. Если ваш диск сейчас виден как **/dev/sda**, запустите

```
udevinfo --attribute-walk --path /block/hda | less
```

Если вы хотите запросить конкретный раздел, допустим, **sda1**, используйте **/block/sda/sda1** как путь. Посмотрев вывод, вы найдете строки вида

```
ATTRS{model}=="model_code"
ATTRS{vendor}=="manufacturer"
```

которые можно использовать для идентификации устройства. Затем отредактируйте или создайте файл **/etc/udev/rules.d/10-local.rules** и добавьте в него строки

```
BUS=="usb", KERNEL=="sd*",
ATTRS{model}=="model_code", ATTRS{vendor}=
="manufacturer", NAME:="extdisk%n",
SYMLINK+="%k"
```

Пункты, использующие **==** – это условия для проверки, которые должны быть удовлетворены, чтобы правило выполнялось. Первые два дают уверенность, что вы работаете с USB-диском или разделом, следующие два скопированы прямо из вывода *udevinfo*.

Следующий пункт задает имя узла **/dev/extdisk** для диска и **/dev/extdiskN** для N-го раздела. Финальная часть создает символьную ссылку на устройство, используя исходное имя, которое действовало бы без этого правила.

Эти правила применимы к диску и любым разделам на нем, но вы также можете перечислить индивидуальные разделы, полезные для устройств типа камер, MP3-плейеров или карт памяти в вашем кардридере. Например

```
BUS=="usb", KERNEL=="sd?[0-9]",
ATTRS{model}=="model_code", ATTRS{vendor}=
="manufacturer", NAME:="camera",
SYMLINK+="%k"
```

В этом случае монтируются только разделы, удовлетворяющие шаблону **sd[a-z][0-9]**, который определяет **sd** и следующую за ним одну букву или цифру; такой раздел появляется как **/dev/camera** вместо **/dev/camera1**. Немало полезной информации о написании правил *udev* находится на www.reactivated.net/udevrules.php.

Может оказаться, что *Gnome Volume Manager* продолжает монтировать устройства с именами вида **/dev/disk-n**. Самое простое решение – давать разделам при создании метки (volume label), чтобы менеджер томов использовал их. Для именования раздела ext3 при его создании, используйте

```
mke2fs -j -L Volume_Name /dev/sda1
```

а для добавления или изменения имени тома в существующей файловой системе –

```
tune2fs Volume_Name /dev/sda1
```

3 О dcop, слайд-шоу и качающихся окнах

Я только что прочел учебник Дэвида Фили KDE: «Галерея на рабочем столе» в **LXF98**. Я нашел ее по-настоящему интересной и, надеюсь, полезной для меня. Сейчас я использую функцию KDE по случайному выводу обоев рабочего стола (Слайд-шоу из случайного набора изображений), и некоторые »


KDE Desktop Leave boring images behind and transform your wallpaper

KDE: Desktop Art

Gallery



Краткая справка про...

Любезность программ

Помните, как мама учила вас вежливости? Она хотела, чтобы вы были дружелюбнее к другим детям и не отнимали у них все игрушки. То же самое применимо к программам, запускаемым на вашем компьютере. У вас конечное число циклов работы процессора, выделяемых на все ваши приложения; если одна программа заберет львиную их долю себе, пострадают другие программы. Linux поставляется с командой, заставляющей программы вести себя любезнее, и она называется *nice*. вы можете запустить ее так

```
nice любаяпрограмма
```

или

```
nice -n 5 любаяпрограмма
```

Число после **-n** – это приращение «любезности» программы. По умолчанию программы запускаются с «любезностью» **0** (чем выше число, тем «любезнее» запускаемая программа); программы уступают программам с меньшей «любезностью», но будут иметь приоритет над программами с большей любезностью. Цифра в команде *nice* добавляет значение к существующему: так, дважды запустив **nice -n 5**, получим любезность, равную **10**.

Наивысшая любезность равна 19, и она специально предназначена для использования процессора, только когда он никому не нужен. Программы, запущенные с таким параметром, могут работать очень медленно, но на реакции системы это никак не скажется. Обычный пользователь может только увеличивать значение любезности, но при наличии доступа root можно также вводить отрицательные значения. Сломать систему очень легко, поэтому «строить» программы, делая их любезнее, разрешается только суперпользователям.

Хотя *nice* помогает в загрузке процессора, она ничего не может поделать с программами, требующими частого доступа к диску и блокирующими процессор ожиданием ввода/вывода. Решением данной проблемы является *ionice*, которая делает со вводом/выводом то же, что *nice* с процессором.

› **Программы могут съедать много процессорного времени, не замедляя при этом других, если вы велите им стать любезнее. top показывает степень любезности, а также позволяет ее менять.**

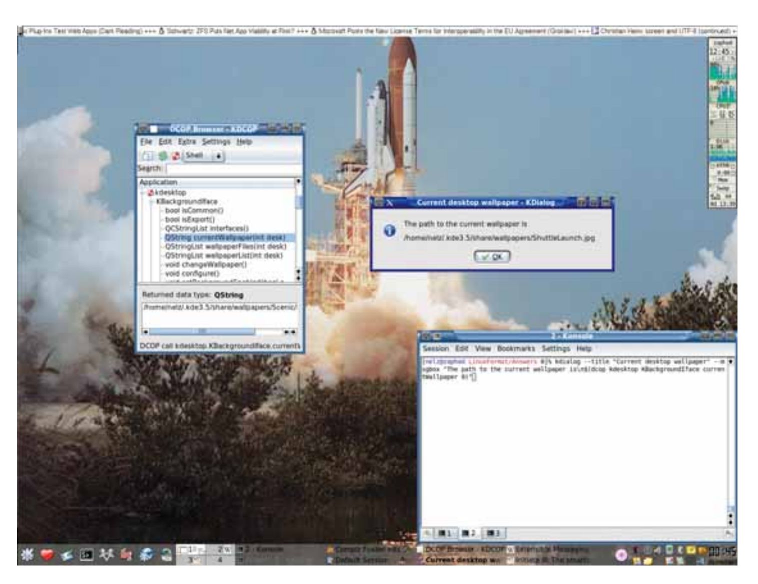
» изображения хотелось бы оставить на экране подольше или скопировать и установить в мой ноутбук или рабочий ПК. Но вот беда: в моей коллекции около 3000 изображений, и найти там желаемое не так-то просто. Мой оконный менеджер – *Beryl*, но я могу вернуться на *kwin*, чтобы это заработало (когда я занимался по учебнику, я так и сделал).

Я заставил первую команду работать на текущем рабочем столе (он у меня только один), но вторая **currentWallpaper** вылетает с сообщением 'object not accessible (объект недоступен)'. По-моему, это из-за слайд-шоу, но толком не знаю.

Я надеялся получить маленький скрипт, который просто использует *kdialog* или что-то похожее и говорит мне имя изображения. Не могли бы вы подсказать, как изменить код из учебника, но чтобы он работал также и с *Beryl*?

Нейл Уильямс [Neil Williams]

Команда точно работает со слайд-шоу: я сам использую ее вариант для удаления текущих обоев (я просто могу свалить все изображения в директорию с обоями, а затем выкинуть несимпатичные, когда они появятся на рабочем столе). Сообщение "object not accessible" у меня воспроизвелось, когда я по ошибке использовал с **currentWallpaper** *KwinInterface* вместо *KBackgroundIface*. По-моему, легче всего экспериментировать с *dcop*, запустив *kdcop*, что позволит обозревать интерфейс и команды *dcop* – можно даже скопировать и вставить строки из окна *kdcop* в оболочку или скрипт. Вот скрипт, который выводит имя текущих обоев.



› DCOP может вернуть имя обоев рабочего стола, даже при слайд-шоу, а *Kdcop* покажет вам, как.

```
#!/bin/sh
IMAGE=$(dcop kdesktop KBackgroundIface currentWallpaper 0)
kdialog --title "Current desktop wallpaper" --msgbox "The path to the current wallpaper is\n${IMAGE}"
```

Даже если у вас только один рабочий стол (мы и не думали, что такое еще бывает – разве что с тоски по Windows), вам необходимо указать его номер. Использование **0** здесь означает «текущий рабочий стол», это сокращает проверку того, на каком рабочем столе мы находимся. Вы можете уместить скрипт в одну строку, используя командную подстановку, то есть назначить команде псевдоним или горячую клавишу без необходимости иметь отдельный сценарий в файле:

```
kdialog --title "Current desktop wallpaper" --msgbox "The path to the current wallpaper is\n$(dcop kdesktop KBackgroundIface currentWallpaper 0)"
```

Так как ваш оконный менеджер *Beryl*, а не *Kwin*, вызовы *kwin* через *dcop* работать не будут, но использованные здесь вызовы *kdesktop* сработают (по крайне мере, они работали с *compiz-fusion*). *Kdcop* точно доложит, что у вас есть.

4 Многоядерные процессоры

Двух- и четырехъядерные процессоры так подешевели, что я заинтересовался, сможет ли Linux воспользоваться этой технологией сейчас или в ближайшем будущем. Просмотрев доступную мне информацию, я так и не понял, какую выгоду они смогут принести мне на компьютере с Linux. Также, существует ли дистрибутив или версия Linux, более подходящий при использовании новых процессоров, чем другие?

Уилсон Е. Стивенс [Wilson E Stevens]

Linux поддерживает несколько процессоров (или процессорных ядер) уже несколько лет: Я пишу эти строки на Core2Duo (хотя редактору *Kate* не нужна вся мощь двух ядер, даже для проверки моего правописания). Большинство дистрибутивов поддерживают несколько процессоров «из коробки» – главное, чтобы ядро было собрано с поддержкой SMP (симметричной мультипроцессорности). Некоторые дистрибутивы включают ее в основное ядро, другие имеют отдельное ядро с SMP, которое выбирается, если установщик определит более одного ядра процессора. Существует несколько способов

Часто задаваемые вопросы...

Магические клавиши

Есть проблемы? Иногда проще прервать процесс и запустить его снова.

» Почему компьютеры имеют кнопку Reset?

Потому что они поставляются с операционной системой от Microsoft.

» Эта шутка так стара, что ее слышали еще динозавры. По-вашему, программы Linux никогда не падают?

Ядро Linux очень стабильно, и крах приложений обычно слабо влияет на состояние системы. Но иногда дефектный процесс может навредить, если захватит все процессорные циклы или всю память (или все вместе) и в итоге потребует перезагрузки. Или случайно запаниковавшее ядро может заморозить процесс загрузки.

» Значит, нам все-таки надо нажимать на Reset?

А вот и нет. Ядро Linux немного умеет колдовать: если вы нажмете Alt + SysRq и командную клавишу, ядро всегда воспримет команду, даже будучи полностью заблокированным. Так как это функция ядра, она готова к работе сразу после его загрузки, и вы можете использовать ее во время запуска.

» Я не смог найти клавишу SysRq, что я делаю не так?

Клавиша SysRq также используется для Print Screen, на ней может быть одна из этих надписей или обе.

» Что это за командные клавиши?

Их несколько, для выполнения различных операций, но чаще всего используются E, I, S, U и B, нажимаемые именно в таком порядке.

- E посылает всем процессам сигнал TERM.
- I посылает сигнал KILL. Не используйте ее на удаленных сессиях, так как она разорвет сетевое соединение.
- S синхронизирует все смонтированные файловые системы, т.е. немедленно записывает на диск все данные из кэша.
- U отмонтирует все смонтированные файловые системы и перемонтирует их в режиме только-для-чтения.

Последние две операции гарантируют, что содержимое ваших дисков корректно запишется и не будет повреждено. Это сделает безопасным нажатие еще одной командной клавиши:

- B для перезагрузки, которая перезагружает систему немедленно.

Так как ваши диски синхронизированы, вы не увидите предупреждений при перезагрузке, и не надо будет запускать *fsck*.

» Это все прекрасно при наличии доступа к клавиатуре компьютера, а если он удаленный и лишен периферии?

В этом случае можно использовать магию команд вроде этой:

```
echo s >/proc/sysrq-trigger
```

Если вы способны просочиться в компьютер, то можно выполнить синхронизацию и выключение.

» Как я могу запретить другим использовать эти клавиши?

Просто набрав:

```
echo 0 >/proc/sys/kernel/sysrq
echo 1 >/proc/sys/kernel/sysrq
```

вы выключите и включите их соответственно; сделать это можно только от имени суперпользователя. вы также можете включить команды выборочно: это описано в документации.

» Какие другие клавиши и команды можно использовать?

Полный список и всякие подробности можно найти в документации ядра. Если у вас установлен исходный код ядра, вы можете найти ее в **/usr/src/linux/Documentation/sysrq.txt**; также посмотрите в сети на **www.mjmwired.net/kernel/Documentation/sysrq.txt**.

легко определить наличие поддержки SPM; вывод команды

```
cat /proc/cpuinfo
```

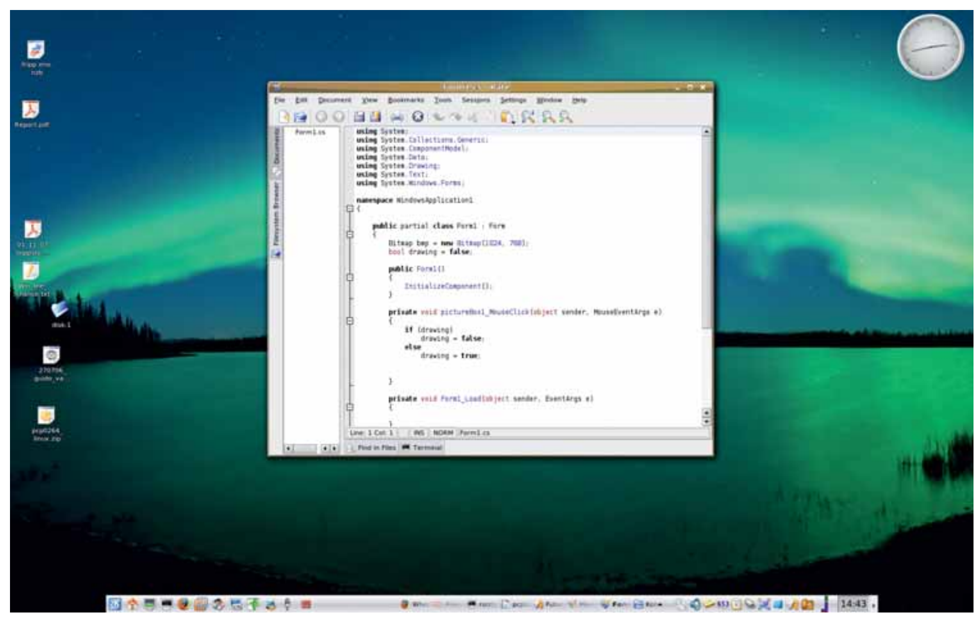
показал вам цифру "cpu cores"? Попробуйте запустить *top* в терминале и нажать 1, при этом вывод нагрузки процессора переключается с показа общей картины на показ индивидуальной загрузки процессоров, и наоборот. Запуск

```
zgrep SMP /proc/config.gz
```

выведет **CONFIG_SMP=y**, если поддержка SMP включена, при условии, что ваше ядро поддерживает **/proc/config.gz**, в противном случае вы увидите невразумительную ошибку.

Поддержка SMP в ядре улучшает многозадачный режим, так как программы могут перекидываться на менее загруженный процессор, но большинство приложений продолжают использовать только один процессор [ОС может сама распределять программы по процессорам, – прим. пер.]. Однако некоторые программы с интенсивной загрузкой процессора могут разделять ее между несколькими ядрами. Например, в *ffmpeg* опция **-threads** делит задачу перекодирования видео между несколькими процессорами, существенно снижая время обработки. При компиляции программ это также полезно, ведь большинство из них состоит из множества мелких файлов, требующих компиляции, а это легко распараллелить. При установке переменной окружения **MAKEOPTS** в профиле пользователя на **-j3** для двухъядерных и **-j5** для четырехъядерных процессоров, программы обычно компилируются гораздо быстрее. Заметим, что используемое здесь число на единицу больше количества процессоров (**-j2** используется для однопроцессорных систем), для обеспечения наибольшей эффективности использования процессора.

Если вы загружаете систему интенсивными задачами вроде компиляции или перекодирования видео, попробуйте использовать команду *nice* (смотрите *Краткая справка про...* на стр. 115) для улучшения отклика вашей системы при работе.



› **Увеличение мощности процессоров не сильно влияет на простые приложения типа *Kate*.**

5 Обновления сбоят

В У меня Compaq Deskpro 650 МГц со 128-МБ ОЗУ и 10 ГБ жестким диском. На нем стоит Mandrake 10.0, который уже устарел, и я пытаюсь обновиться до – или установить с нуля – Mandriva 2007. Компьютер постоянно слетает в процессе установки. Иногда установка удается, но при попытке загрузки системы она всегда зависает. Я вижу на экране много текста, что наводит на мысль о проблемах с ядром. Пробовала другие дистрибутивы – Fedora, Ubuntu, Slackware и Gentoo – и везде та же проблема. Есть ли способ установить новые дистрибутивы (Mandrake 10.1 – и тот отказал) и заставить их работать?

Райнхильде Вермейлен [Reinhilde Vermeulen]

О Тот факт, что ваши проблемы с разными дистрибутивами не всегда проявляются в том же месте, говорит, что проблема может исходить от оборудования – вероятнее всего, сбой памяти, перегрев или плохое качество блока питания. Прежде чем этим заняться, рекомендую запустить инсталлятор в текстовом режиме. Графические инсталляторы требуют много памяти, гораздо больше, чем нужно для запуска установленного дистрибутива: они записывают все необходимое на большой ram-диск, чтобы все было доступным, когда вы смените CD. Установка в текстовом режиме существенно снижает требования к памяти.

Если это не поможет, вам нужно проверить вышеупомянутые проблемы. Тестирование памяти делается просто: большинство установочных дисков дистрибутивов включают *memtest* как опцию в меню. Выберите ее и дайте ей поработать как можно дольше: понадобится как минимум два прохода, и лучше запустить ее на всю ночь. Перегрев возможен при поломке вентилятора или засоре (извините за выражение) вентиляторов, радиаторов или вентиляционных отверстий. Для очистки от пыли их продувают сжатым воздухом. Неисправный блок питания также может вызывать случайные перезагрузки и зависания, но единственный способ протестировать его – это замена.

Если все это не сработает, обратите внимание на последний десяток строк текста на экране при сбое. Часть этих сообщений идет от ядра, а часть от различных программ, запускаемых при загрузке системы. Знание содержания этих сообщений может помочь в определении проблемы. Если сообщения бессвязны, у вас точно сбой в оборудовании.

6 Пропали файлы?

В У меня не устанавливаются никакие программы: не могу выполнить инструкции в журнале, сколько ни пытаюсь. Конечно, архивы я распаковываю, это не проблема. К примеру, я пытался установить *Digikam*, и когда я запускал configure, все проходило на ура, почти до конца. Тут я получал сообщение об ошибке: программа не могла найти важный файл *kde-config* и спрашивала, правильно ли у меня установлен KDE.

Debian был загружен с официального DVD, и инсталяция работала как часы. Установка-то прошла хорошо, но я явно не способен уйти от рабочего стола. Эта машина не подключена к интернету. Буду очень признателен, если вы мне скажете, где я могу или должен искать *kde-conf*. Я не смог его найти нигде.

Билл Брукс [Bill Brooks]

О Во-первых, инструкции в журнале по установке из исходных текстов – это общие рекомендации. Они годятся в более чем 90 процентов случаев, но конкретные инструкции помещаются внутри архивов с исходниками, в файлах README и/или INSTALL. Обычная проблема пользователей Debian и систем на основе RPM в том, что большинство пакетов разбито на два: стандартный *foo-x.y.z* и *foo-devel-x.y.z*. Обычно нужно устанавливать только *foo*, но второй содержит заголовочные файлы, необходимые для компиляции другого ПО, работающего с этой программой. То есть, когда вы видите сообщение об ошибке вида "libfoo not found" и знаете что *foo* установлен, это значит, что вам нужно установить еще и пакет *foo-devel*.

В вашем случае этого может не быть: *kde-config* – это часть *kdelibs*. Запустите команду **which kde-config** в терминале, она должна вернуть путь к программе (**/usr/bin/kde-config**). Если этого не произошло, проверьте, существует ли **/usr/bin/kde-config**. Если файл на месте, но which не может его найти, значит, что-то неправильно в установках **$PATH**. Вывод **echo $PATH** должен включать **/usr/bin**, если этого нет, проверьте установки в **/etc/profile**.

Если *kde-config* там нет, возможно, вам надо переустановить содержащий его пакет. Так и слышу ваш вопрос: «Откуда я знаю, какой пакет мне нужен?» Debian держит мощный движок поиска пакетов на сайте http://packages.debian.org. Зайдите в »

› **Движок поиска пакетов Debian – хороший способ узнать, какой пакет содержит нужный файл.**

» Search the contents of packages, введите **kde-config** и выберите Packages that contain files named like this; он должен вернуть вам, что искомый содержится в пакете *kdelibs4c2a*, который должен быть у вас уже установлен, если вы используете программы KDE. Переустановите пакет и *kdelibs4-dev*, потому что это вам понадобится, и проверьте, что файлы теперь на своих местах.

7 Проигрывая DVD

Моя работа требует поездок по всему миру, и в свободное время, как и многие другие люди, я развлекаюсь, смотря фильмы на DVD. Я запускаю Fedora 7 на моем T60 Thinkpad с установленным *VLC* и всеми расширениями из репозитория *Livna*, но могу смотреть только незашифрованные DVD. У меня установлен *libdvdcss*, и все равно я не могу смотреть большинство фильмов, которые мне попадаются. Я слышал, что привод Matshita DVD не может проигрывать зашифрованные диски. Это правда? Если нет, то есть ли ПО для излечения этой болезни?

Петер Берг [Peter Berg]

Ваш привод ориентирован на определенный регион и позволяет проигрывать зашифрованные диски только для него. Консорциум DVD разделил мир на шесть регионов, и привод может проигрывать только диски того региона, на который он настроен. Регионы пронумерованы так:

1. **Северная Америка (США и Канада)**
2. **Европа, Ближний Восток, Южная Африка и Япония**
3. **Юго-Восточная Азия, Тайвань, Корея**
4. **Латинская Америка, Австралия, Новая Зеландия**
5. **Бывший Советский Союз (Россия, Украина и т.д.), остальная Африка, Индия**
6. **Китай**

В большинстве приводов можно сменить регион, используя нечто вроде *regionset* (**http://linvdr.org/projects/regionset**), но это ловушка. Многие приводы позволяют сменить регион только четыре-пять раз, а затем блокируются на последнем установленном регионе, от чего мало пользы, если вы ездите из региона в регион, покупая DVD по дороге. На ноутбуке вы даже не сможете сменить привод на регионнезависимый. На некоторых приводах можно обновить прошивку, что позволит менять регионы бесконечно или сделать привод работающим со всеми регионами. К сожалению, это не относится ко всем приводам Matshita, и вам может не повезти. Информация о многих приводах имеется на **www.rpc1.org**.

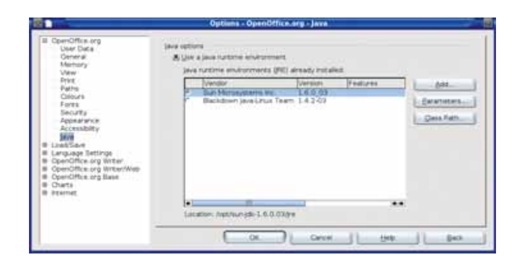
Если на вашем приводе нельзя обновить прошивку (у некоторых приводов прошивка зашифрована), остается вариант перекопировать ваш DVD в другой формат на другом компьютере с регионнезависимым приводом. Этот метод хорошо работает и спасает вас от необходимости тащить с собой DVD, если вы запишете снятые копии на жестком диске, но не поможет проигрывать DVD, купленные за границей. *dvd::rip* (**www.exit1.org/dvdrip**) – хороший выбор для этого.

8 OpenOffice.org и Java

Я установил *OpenOffice.org 2.3* с LXF99 по инструкциям в журнале. Большинство его частей работают нормально, но когда я пытаюсь открыть файлы *Base*, он требует обновить среду Java, которая, похоже, исчезла. Я скачал и переустановил Java с Sun, но так и не получил доступ к файлам моей базы данных – получаю сообщение, спрашивающее меня о расположении Java с помощью Tools > Options [Сервис > Параметры], но не могу обнаружить ни одной установки Java.

Дональд Кэмбелл [Donald Campbell]

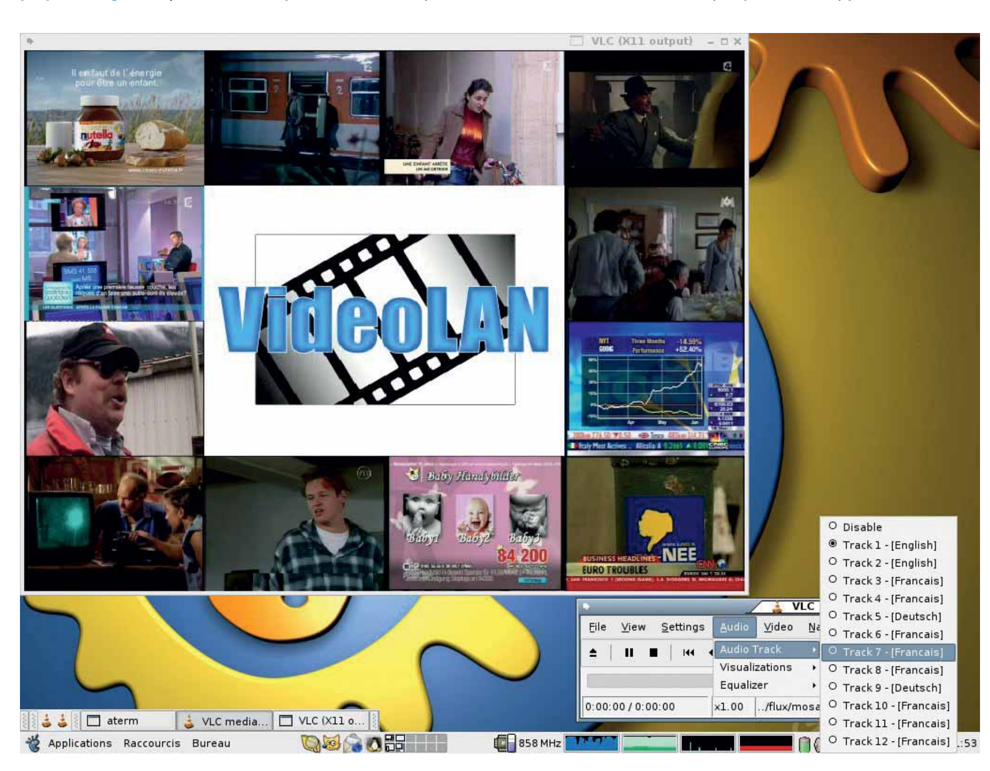
OpenOffice.org нуждается в Java для базы данных и справочной системы, но остальная часть пакета будет работать и без нее. Лучший способ установить Java – через менеджер пакетов вашего дистрибутива. Загруженная с Sun версия будет работать, но старайтесь по возможности получать ПО через менеджер пакетов, это снижает вероятность конфликтов и проблем с зависимостями в будущем.



› **OpenOffice.org может нуждаться в подсказке, какое окружение Java использовать.**

Заставить *OpenOffice.org* работать с Java проще некуда. Выбираем Options [Параметры] из меню Tool [Сервис] и переходим к OpenOffice.org >Java. Включаем Java, отметив квадратик Use a Java runtime environment [Использовать виртуальную машину Java], и ждем, пока Ваша система просканируется на предмет наличия установок Java, на что может уйти от нескольких секунд до минуты, в зависимости от скорости вашей системы. Так как это Java, то за это время вы успеете выпить чашечку кофе. В итоге вы должны увидеть список установленных Java Runtime Environment (JRE); выберите одну и нажмите OK. Придется выйти из *OpenOffice.org* и перезапустить его, чтобы изменения вступили в силу.

Если *OpenOffice.org* не может найти ваш JRE, можете нажать кнопку Add (Добавить) и указать путь к нему вручную: нечто вроде **/opt/sun-jdk-1.6.0.03/jre/bin**. Если вы устанавливали Java через менеджер пакетов вашего дистрибутива (а не менеджер пакетов *OpenOffice.org*), можете использовать его, чтобы посмотреть содержимое пакета, оно подскажет вам, где все установилось.



› ***VLC*** **воспроизводит огромное число форматов, включая DVD, но DVD-привод вашего компьютера может быть «прикован» к проигрыванию дисков только одного региона. Вот такая вот «свобода выбора».**



Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, возможно, это потому, что мы уже ответили на похожий вопрос.

Решение некоторых проблем с USB



Нет длинных имен

В Я использую USB-брелок на компьютере с Windows и скачиваю на него файлы из интернета. Затем подключаю брелок к компьютеру с Linux дома и нахожу, что все файлы с длинными именами переименовались с использованием "~" Например, **mylongfile3.mp3** переименовался в **mylong~1.mp3**.

Если бы вы смогли помочь мне с этим, я был бы очень признателен, так как понятия не имею, что тут неправильно. Сперва я использовал grml.org – дистрибутив на основе Debian, но нашел, что проблема имеет место и на других дистрибутивах, например, Knoppix (тоже на базе Debian).

Джорди Барнс [Jordi Barnes]

О Это похоже на проблему с опциями, используемыми при монтировании файловой системы USB-брелка. По умолчанию на этих устройствах используется FAT16 или FAT32, которые не поддерживают длинные имена напрямую и поэтому используют клуджи (да, знаю: то, что Windows использует клуджи, может шокировать) для отображения видимых вами коротких имен в соответствующие корректные длинные. Файловая система vfat в ядре Linux поддерживает это, а msdos – нет. Запустите команду **mount** в терминале, и вы увидите строки типа этих для вашего USB-брелка.

```
/dev/sda1 on /media/usbstick type
msdos (rw)
```

Если тип показывается как **msdos**, корень проблемы найден – теперь вам надо добиться, чтобы ваш брелок монтировался корректно. Если вы используете записи в **/etc/fstab** для монтирования, измените тип файловой системы, третий параметр в строке, с **msdos** на **vfat** или **auto**. Если вы используете систему автоматического монтирования вашего дистрибутива, то тип файловой системы должен идентифицироваться правильно.

Происходит ли Ваша беда со всеми брелками, которые вы пробуете, или только с этим? Если только с ним, то, наверное, у вас есть повреждения файловой системы, из-за которых команда mount неверно определяет файловую систему. Существуют обходные способы, но лучшим решением будет переформатирование брелка для устранения повреждения (или его замена, если это не поможет). Я не смог воспроизвести эту проблему на **grml.org**, поэтому подозреваю, что ваш USB-брелок поврежден или сломан. Память, используемая в этих устройствах, имеет ограниченный срок жизни, и, возможно, он уже вышел.

Загрузка с USB

В У меня ноутбук Pentium 3, 866 МГц с жестким диском на 20 ГБ и Mandriva 2005LE, с загрузкой через *Grub*. Я хочу установить Ubuntu 7.04 на внешний жесткий диск, сделав двойную загрузку. Я знаю, что доступ к внешнему диску USB2 будет медленным, так как моя машина имеет только порты USB1.1.

Моя проблема в том, что BIOS ноутбука не может сделать USB-диск первым загрузочным устройством. Как модифицировать загрузчик для доступа к USB-диску или использовать дисковод для установки загрузчика? Я испытал трудности при поиске соответствующей информации по этому вопросу, но может быть эта тема уже обсуждалась в LXF...

shoreacre, из форума

О Для загрузки с USB-диска вы можете использовать имеющийся загрузчик. Для этого хорошо использовать альтернативный установочный CD Ubuntu, он дает больше контроля над параметрами установки.

Установите Ubuntu на внешний диск обычным путем, но не разрешайте записать загрузчик в MBR, как обычно делается. Вместо этого установите *Grub* в корневой раздел – это, скорее всего, **/dev/sdb1**, если внешний диск – **/dev/sda**. Затем загрузитесь в Mandriva как обычно и смонтируйте раздел с Ubuntu. Mandriva 2005 использует **/dev/hd*** для IDE-дисков, так что ваш USB-диск теперь должен быть в **/dev/sda**. Откройте терминал суперпользователя и запустите

```
mkdir /mnt/tmp
mount /dev/sda1 /mnt/tmp
ВашТекстовыйРедактор /boot/Grub/
menu.lst /mnt/tmp/boot/Grub/menu.lst
```

для загрузки обеих конфигураций загрузчиков в ваш любимый текстовый редактор. Теперь у вас есть выбор: быстрый и простой метод или вариант поприятнее-глазу-но-позаковыристей-в-установке. Для «быстрого и простого» добавьте следующие строчки в конец файла **menu.lst** Mandriva:

```
title Ubuntu menu
root (hd1,0)
chainloader +1
```

Эти строки добавят в меню запись, запускающую загрузчик Ubuntu. Чтобы все делать из одного меню, вам придется скопировать пункты меню загрузчика Ubuntu в меню Mandriva. Посмотрите на его основные опции, которые идут первыми. Они содержат три строчки: заголовок, показываемый в меню, ядро для загрузки с набором опций и используемый **initrd**. Скопируйте эти строки в **menu.lst** Mandriva и исправьте заголовок на что-нибудь более понятное, затем измените пути к устройствам для корректного их определения. *Grub* считает от нуля, так что **(hd1,0)** – это первый раздел **(0)** на втором диске. вы можете просто включить этот путь в строки **kernel** и **initrd**, или (как я предпочитаю) как отдельную запись root.

Содержимое вашего **menu.lst** должно выглядеть так (для Ubuntu 7.10):

```
title Ubuntu 7.10
root(hd1,0)
kernel /boot/vmlinuz-2.6.22-14
root=/dev/sda1 quiet splash
initrd /boot/initrd.img-2.6.22-14
```

вы также должны скопировать другие пункты меню Ubuntu сходным образом, или оставить как было, для тех редких случаев, когда вам нужна только система по умолчанию.

Если при выборе этого меню происходит ошибка File not found, значит, вы неверно задали путь. Порядок дисков задается в настройках BIOS, но даже после этого *Grub* не сможет загрузиться с USB на всех системах. Проверьте корректный путь к ядру, запустив *grub* в терминале суперпользователя (или нажав **C** в меню *Grub*) для входа в оболочку *Grub*. Затем запустите

```
find /boot/vmlinuz-2.6.22-14
```

Команда должна показать корректный путь к ядру, включая номер диска. Убедитесь, что он совпадает с командой root в вашем меню. LXF

Лучшие новинки открытого ПО на планете

LXF HotPicks



Грэм Моррисон

Ну откуда он знает все, что происходит в мире свободного ПО?! Тут явно не обошлось без двойного зрения, магии вуду и гадания на кофейной гуще...

В ЭТОТ РАЗ ТОЛЬКО ДЛЯ ВАС: Gnome-Inform » Yakuake » Motion » Project-M » Zim » BattleTanks » Bos Wars » Synkron » phpMyAdmin » TaskJuggler

Интерактивная среда разработки

Gnome-Inform

Версия 0.5/4X60 **Сайт** http://sourceforge.net/projects/gnome-inform7

Хотя на www.fightingfantasygamebooks.com все еще попадаются книги-игры, написанные подражателями Стива Джексона или Иэна Ливингстона, до настоящей игры им далеко – это сродни предложению винной жевательной резинки алкоголику. О чем я? Кончина интерактивной фантастики как игрового жанра тяжестью легла на мое сердце. В 1980-х годах эти игры сияли, освещая плоский двумерный мир пиксельной графики и сюжет в стиле вверх/вниз/влево/вправо/огонь: интерактивная фантастика ограничивалась только воображением игрока. Классная работа в этой области приравнивалась к хорошей книге: даже реальные авторы ввязались в драчку, вместе с Дугласом Адамсом помогая Infocom создавать его лучшие игры и сочетая специфический юмор и цинизм Адамса с механикой головоломок Infocom. На какое-то короткое время вы становились Артуром Дентом. Поставка даже включала миниатюрный космический флот в качестве доказательства. Если вы никогда не играли в эти игры, обязательно сыграйте. Если вы не играли в них лет двадцать, сейчас настала пора попробовать их снова. Но жанру действительно требуется новый творческий талант. Для записи собственной интерактивной фантастики как раз и нужен *Gnome Inform 7*: это интегрированная среда разработки вашей собственной Вселенной, включающая все инструменты для ее создания с максимальной легкостью. Основное окно редактирования, с выделением синтаксиса, дополнено хорошей документацией и дополнительными инструментами, чтобы помочь вам сконструировать и организовать ваше творение. Не позволяйте себя отпугнуть от этого удовольствия из-за его программистского уклона.

Интерфейс Gnome-Inform

Выпуск вашей игры
По завершении, нажмите на эту кнопку для создания файла, который можно проигрывать в любом приложении Inform.

Журнал ошибок
Как и в программировании, здесь случаются ошибки. Журнал ошибок постарается вам помочь.

Отличная документация
Как и следует ожидать от приложения, написанного людьми, которые по жизни писатели, документация на высоте.

Разделение окна
Вы можете просматривать два разных кода в главном окне, экономя свободное места.

Просмотр игры
Нормальный вид включает код игры слева и запущенную игру справа.

Индекс объектов
Все, что вы можете добавить в игру, используя естественный синтаксис, записывается и сортируется в индексе игры.

Всякая всячина

Использовать Gnome Inform 7 очень легко и приятно. Его основа – язык Inform, используемый для построения игры. Он не похож на обычный язык программирования, так как использует естественный язык для описания взаимоотношений между объектами и объектов внутри игры. Например, наберите «стакан вина в туалете», и Inform поймет, что «стакан вина» – это объект, находящийся в комнате под названием «Туалет». Наберите «Туалет – к югу от Офиса» и Inform примет к сведению взаиморасположение этих помещений. Добавьте описание для каждого объекта и места в двойных кавычках, и все готово для начала игры. Нажмите кнопку «начать», и игра скомпилируется и запустится в панели справа. Если есть ошибки, то вы получите их длинное текстовое описание. Если нет, можете сразу начать играть в только что созданную игру. Можно изменить исходный код в левой панели, нажать кнопку «начать» на панели инструментов и тут же посмотреть эти изменения в игре. Gnome Inform 7 вернет вас в то же место, в том же состоянии. Это сильно облегчает разработку интерактивной фантастики, особенно если вы знакомы с однообразием процесса «исправить, пересобрать и запустить», типичного для разработки приложений.

Единственная реальная трудность заключается в том, чтобы вникнуть в синтаксис естественного языка Inform, а также в определение «вещи». Поможет страница Индекс [Index]: как вы, видимо, догадались, она генерирует список всех возможных видов объектов, которые можно создать; этот список будет автоматически обновляться, вбирая используемые в текущий момент в игре предметы. Индекс включает также прямые ссылки на документацию для каждого типа объекта (он называется “kind”), и вы в нем быстро освоитесь.

Консоль нового поколения

Yakuake

Версия 2.8 **Сайт** http://extragear.kde.org/apps/yakuake

Если вы были на одной из конференций KDE в этом году, то не могли не заметить, что все разработчики использовали терминал, который волшебным образом раскатывался от верхней границы экрана. Он известен как «терминал в стиле Quake», поскольку происходит из этой игры и ее встроенной консоли, где впервые были применены плавные прокрутки во время появления и ухода терминала с экрана. А KDE-версия, которая была у всех разработчиков, называется *Yakuake* – очередное поколение *Konsole* от KDE. Нажмите некую комбинацию клавиш, и *Yakuake* плавно раскроется перед вами на экране, готовый для ввода любых команд. Нажмите ту же комбинацию клавиш снова, и панель ускользнет из вида. Самое главное, что ваша сессия в терминале продолжает работу. Каждый раз, когда вы запускаете консоль, вы начинаете работу с того же места. Это значит, что в *Yakuake* можно, например, запустить компиляцию программы и свернуть окно терминала на время, пока идет сборка. Вот вам и причина популярности *Yakuake* среди разработчиков – или то, что легко упрятать IRC-чат в консоли. *Yakuake* идентична по функциям *Konsole* и даже использует ту же Schema для изменения внешнего вида, кроме того, можно скачать различные «скины». Вы можете задать «жилплощадь» *Yakuake* на экране (в процентах от свободного места) и часть экрана, откуда она будет всплывать, и, как и в *Konsole*, можно открыть более одной сессии, добавив вкладки. В общем, идеальная замена для устаревающей консоли, и хотя вначале вы, возможно, заподозрите, что гладкость прокрутки – это какая-то уловка, плавное появление и исчезновение консоли с экрана гораздо продуктивнее, чем многократный запуск новой *Yakuake*. Те, кто использует командную строку Linux каждый день, как мы здесь, в **LXF**, несомненно, скоро объявит друзьям и коллегам, что *Yakuake* трудно переоценить.



› **Даже поклонники работы с текстом будут покорены!**

> **«Запустите компиляцию в терминале и уберите его с глаз до ее завершения.»**

Программный датчик движения

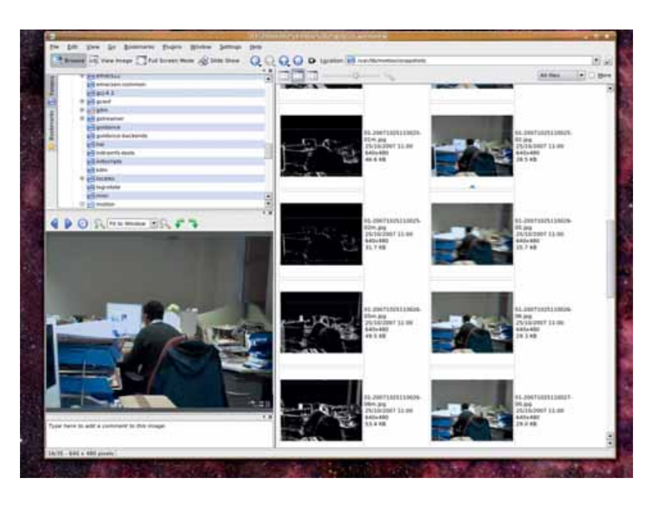
Motion

Версия 3.2.8 **Сайт** http://tinyurl.com/4qhyb

Мы наткнулись на *Motion* при подготовке прошлого выпуска. Прямо и не знаем, почему мы до сих пор не засекли эту замечательную маленькую утилитку. *Motion* определяет движение с вашей камеры и сохраняет соответствующие изображения в папку по умолчанию. Потом вы можете проглядеть эти изображения в поисках того, что вызвало срабатывание. *Motion* запускается из командной строки, но вы этим не смущайтесь.

Набрав **motion** в командной строке, вы увидите настройки, и это полезно само по себе. Мы нашли, что настройки по умолчанию отлично подходят для помещения с переменным освещением и контрастом: алгоритм сумел отследить движение даже при помехах на заднем плане. *Motion* подходит главным образом для использования в самодельных системах безопасности, но вы сразу же захотите изменить место папки для хранения изображений по умолчанию. Лучше всего для этой цели подходит ftp-сервер: даже если жесткий диск вашего компьютера будет украден, изображения сохранятся на удаленном компьютере. Вдобавок программа предлагает дюжину других настроек. Одна из наших любимых – преобразование сохраненных изображений в видеофайл (при помощи FFMPEG): просматривать его гораздо удобнее, чем скакать вдоль сотни отдельных файлов. Можно также обработать изображения, выделив область перехода черным цветом или рамкой. Кроме того, в программу включен потоковый сервер для просмотра изображения с web-камеры в любом браузере. Хотя *Motion* – приложение командной строки, функций здесь хватает, а качество захвата движения не уступает лучшим решениям. Если вы обдумываете, куда девать старые ПК и web-камеру, *Motion* – хороший повод не спешить их выбросить: утилите не нужен мощный процессор, она проста в использовании, и все, что вам понадобится – это набрать **motion** в командной строке.

Единственный недостаток, способный вас оттолкнуть – хроническое отсутствие поддержки многих web-камер под Linux, но это уж не вина *Motion*. Кому нужны драйвера для web-камеры, на сайте Мишеля Ксаарда http://mxhaard.free.fr вы найдете их более 230. Мишель, французский доктор и поклонник Linux, предоставляет ссылки на бинарные пакеты драйверов в форматах **.rpm** и **.deb**.



› ***Motion* создает рамки вокруг областей, где обнаружено движение. Ловко придумано!**

> **«Функций множество, а качество обнаружения движения превосходное.»**

Аудиовизуализация

Project M



Версия 1.01 **Сайт** http://projectm.sourceforge.net

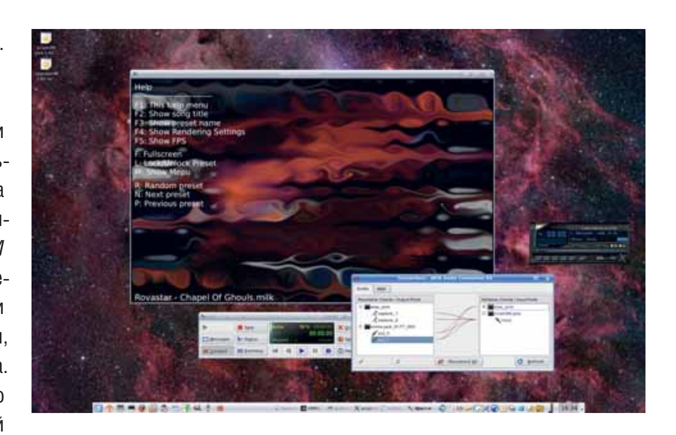
Документация к *Project M* практически отсутствует, и без помощи пары снимков экрана вы нипочем не догадаетесь, что именно делает эта программа. И это несправедливо, потому что она здорово умеет визуализировать музыку. Следуя давним традициям демо-кодеров и сторонников *XMMS*, *Project M* создает графику, форма которой изменяется по мере воспроизведения плей-листа. Придумано такое уже давно, но *Project M* – лучшее, что мы встречали под Linux. Она подобна *MilkDrop*, визуализатору под Windows. *Project M* даже может использовать предустановки **.milk**, а это целая библиотека возможностей. Спасибо разработчикам, которые вывели *Project M* из комы, продлившейся несколько месяцев, и дорастили версию 0.99 до 1.0.

Project M – не самостоятельное приложение: оно работает только в паре с музыкальным ПО. Имеется, например, модуль расширения для использования с XMMS, но мы рассмотрим версию, совместимую с *Jack*. Если вы захотите скомпилировать ее сами, убедитесь, что у вас установлена *cmake*, а также файлы разработчиков для *FRGL* и *GLEW*. Под Ubuntu вам также понадобится опция **-i** *cmake*, чтобы определить **/usr** как префикс для установки. Выбор **/usr/local** по умолчанию не работает, так как распаковщик не находит необходимую библиотеку. Впрочем, авось к моменту прочтения вами этой статьи какая-нибудь добрая душа создаст двоичный пакет – тогда не нужно будет следовать руководству по установке.

Jack вашу так

Связавшись с *Jack*-версией, мы должны были сначала запустить аудиосервер *Jack* с использованием утилиты *Jack Control*: она включена во все дистрибутивы, которые мы сумели припомнить. Затем запускается клиент *Project-M Jack*, который мгновенно появится в менеджере соединений *Jack*. Если вы не использовали *Jack* до этого, то можете немного запутаться, но на самом деле программа довольно удобна. *Project-M* надо подключить к выходу вашего аудио-проигрывателя. Им может быть любой продукт для Linux с выходным модулем *xine-jack* – например, редактор звуковых потоков *Audacity* или программный синтезатор. Как только программа издаст звук, *Project-M* примется показывать свой ряд визуализаций. Нажмите F1 для вызова меню помощи, там вы найдете сочетания клавиш и команды для управления *Project-M*.

Для навигации по пресетам визуализаций используйте клавиши N (вперед) и P (назад) – картинка будет фантастически меняться; можно отображать имя пресета, а если вы используете модуль *XMMS* – то и название проигрываемой композиции. Процесс отрисовки хорошо контролируется, это может быть полезно для слабых карт. Нечего и говорить, что вам понадобится относительно современная видеокарта с ускорением OpenGL (хотя я все-таки сказал это), но не уж очень навороченная.

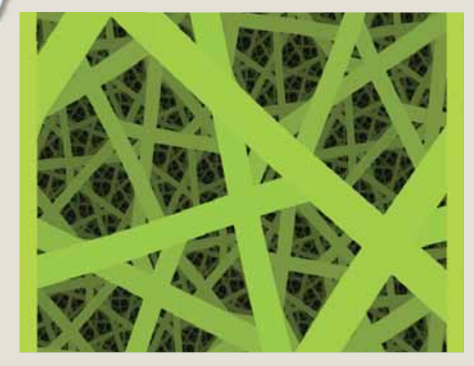


› Используйте *Jack* для подключения вашего аудиопотока к клиенту *Project-M*. На этом рисунке в визуализации нашей музыки участвует модуль *Jack* для *XMMS*.

Если *Compiz* на вашем компьютере идет без проблем, то и с *Project M* проблем не будет.

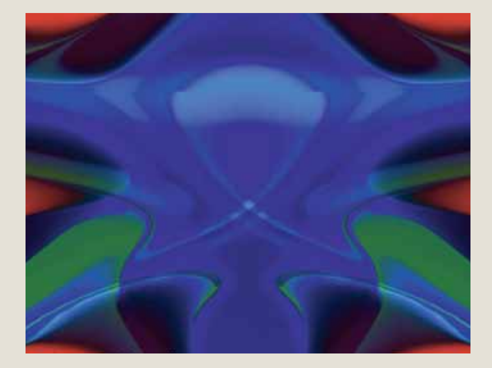
Для тех, кто часто слушает музыку на компьютере, *Project M* – лучшая программа визуализации, с которой нам доводилось работать под Linux. Она очень гибкая в использовании и совместима с библиотекой **.milk**-файлов, так что у вас не возникнет недостатка в пресетах. Может, и легкомысленно тратить ресурсы процессора на музыкальную визуализацию, но выглядит она, несомненно, потрясающе. И если к вам забредут гости, она сумеет отвлечь их от программы вечерних новостей.

Выбранные из множества: Наши любимые пресеты



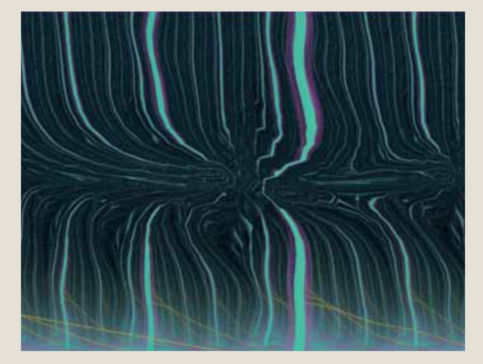
» Telek

Чтобы оценить всю необычность, это нужно видеть в движении: трубы на заднем и на переднем планах перемещаются синхронно.



» Potion of Spirits

Подобно какому-нибудь древнему ритуалу американских индейцев, это настоящий шедевр психоделики, хотя мы побоялись бы оказаться с ним наедине.



» They're so cute

Картинка не отражает всей красоты происходящего на экране. Плавающие полосы атмосферных помех в ритме музыки... Великолепно!

HotGames Развлекательные приложения

Многопользовательская битва

Battle Tanks

Версия 0.6 **Сайт** http://btanks.sourceforge.net/blog

Лучшее в *Battle Tanks* – то, что вам ничего не придется учить.

Запустите игру и громите все подряд – вот и все, что от вас требуется. Цель – одна-разъединственная: истребить противника. Сыграйте на одной из десятков многообразных карт против реальных людей из вашей локальной сети или против компьютера. Также есть вариант игры в союзе, когда вы скопом наваливаетесь на беднягу-компьютера. Вид сверху показывает территорию игры; можно использовать клавиши курсора для навигации по карте и Ctrl для стрельбы из главного оружия, а Alt назначено для второстепенного (в основном это закладывание мин). Предлагается несколько разных видов оружия на выбор, с различными способами стрельбы: от реактивных снарядов и бомб-бумерангов до трехствольных пушек. Найдутся даже орудийные башни, их можно настроить на автоматический обстрел противника. Игра заключается в маниакальном хватании всего подворачивающегося под руку оружия, пока не началась смертельная схватка. Вы можете выйти из своего танка и вскочить обратно, если потребуется, кроме того, многие карты настолько велики, что можно кататься на танке довольно долго, никого не встречая. Это, впрочем, не значит, что можно прекратить пальбу, поскольку кругом носятся такси, машины и автобусы, а также странный вертолет. Когда вы кружитесь по карте и даете промахи, графика обваливается и взрывается; в иллюстрациях много юмора, равно как и умения. Это одна из приятнейших глазу открытых игр среди того, что нам доводилось видеть: благодаря OpenGL-ускорению объекты не просто перемещаются – они струятся, как шелк.

> «Благодаря OpenGL-ускорению движения гладкие, как шелк.»



› **В *Battle Tanks* лучше всего играть простив реальных людей. Тогда и ваши атаки будут иметь реальную подоплеку!**

Нет выхода в сеть? Зато опция раздвоения экрана, и к вам может присоединиться, а точнее – напасть, сосед-коллега. Единственное, что нам хотелось бы еще увидеть, это некую форму поиска в Интернете, чтобы подбирать партнеров тем, кому не с кем поиграть.

Стратегия в реальном времени

Bos Wars

Версия 2.4.1 **Сайт** www.boswars.org

Bos Wars напоминает о золотом веке старых стратегий в реальном времени и очень схожа с *Dune 2*, только производство спайса заменили на сбор тепла раскаленной магмы. Знакомыми кажутся даже технободобные звуки, издаваемые объектами, хотя они и не совсем такие, как утвердительное «Yes, SIR» из *Dune*. Разработчики *Bos Wars* делали игру, явно вдохновившись веcтвудовской классикой. Щелкнув на объекте, вы услышите в ответ вызывающее ностальгию электронное «Yes», а интерфейс типа наведи-и-пали для управления войсками на поле битвы – родной до боли.

Но описывать *Bos Wars* просто как клон *Dune* было бы ошибкой: этой игре есть чем зацепить внимание. Ландшафт, например, стал играть более важную роль: ваша стратегия определяется количеством дорог или границ с соседями. В *Bos Wars* также используется динамическая экономика, зависящая от темпа: другими словами, движение потребляет энергию, добываемую из магмы, и сохраненный запас энергии вас выручит, когда вы угодите в бесплодный угол на карте.

Даешь разнообразие!

Другая особенность, отличающая *Bos Wars* от прочих открытых игр – превосходный искусственный интеллект игрока-компьютера. Вы можете с удовольствием играть против компьютера, с друзьями или со всеми вместе. Игра в сети, возможно, самый стоящий аспект *Bos Wars*, но вы можете отточить свою стратегию, сыграв против компьютера; после этого вам будет легче раздолбать коллег. Если вы раньше не играли в подобного рода игры, вам пригодится обучающая миссия для ознакомления с процессом, а когда схема игры вам надоест, можете воспользоваться редактором уровней для создания собственной миссии. В *Bos Wars* хочется играть снова и снова.

> «В Bos Wars интеллект игрока-компьютера превосходен.»



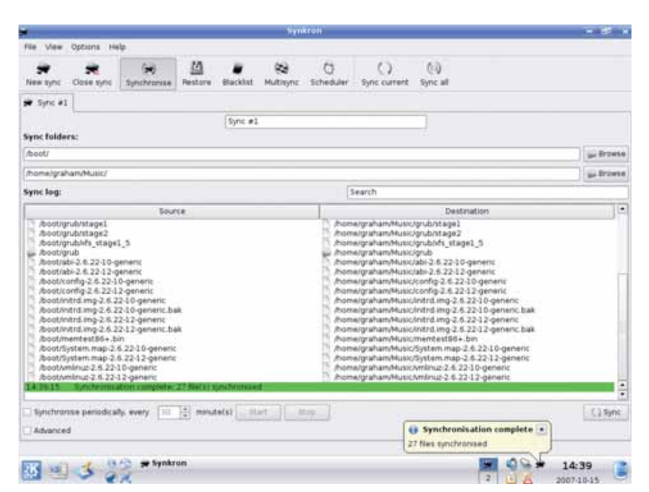
› **Отличный обучающий режим *Bos Wars* помогает одолеть основы игры.**

Синхронизатор файлов

Synkron

Версия 1.01 **Сайт** http://synkron.sourceforge.net/

Несмотря на то, что чуть ли не у каждого свой MP3-плейер, на свете довольно мало программ, позволяющих вам синхронизировать с ним свою музыкальную коллекцию. *Synkron* не разрабатывался специально под эту задачу, но именно для нее он и хорош. *Synkron* просто синхронизирует содержимое одной папки с другой, подобно команде **rsync**, с тем преимуществом, что его пользовательский интерфейс доступен пониманию, и вы в любой момент точно видите, что происходит, тогда как *rsync* оставляет вас в неведении. Часть названия программы «kron» намекает, что *Synkron* способен выполнять автоматическую синхронизацию по заданному расписанию, позволяя вам, например, синхронизировать вашу музыкальную коллекцию каждый вечер, или периодически делать резервную копию рабочей папки – хоть каждые 15 минут, если надо. Список функций дополнен черным списком: при синхронизации часть файлов можно игнорировать. Перечисленные функции покрывают практически все нужды при использовании подобного приложения, и мы сочли *Qt*-интерфейс довольно гибким. Например, несколько заданий могут выполняться одновременно, и каждое получит свою вкладку в главном окне. Прогресс каждого задания отслеживаться на дисплее обновления статуса. Это важно, потому что завершение задания часто требует достаточно много времени, в зависимости от количества файлов и папок, выбранных для синхронизации. И все же *Synkron* – лучший пример в своем роде, из нами виденных, особенно по части синхронизации музыки. Хорошо бы кто-нибудь создал такое же приложение специально для музыкальных файлов, чтоб не проверять каждый раз, какие файлы находятся в обеих папках (iPod использует для этих целей базу данных). *Synkron* легко модифицировать нужным образом, и если у кого-то найдется свободное время на вклад в развитие Open Source, вот ему и занятие.



› **Можно свернуть главное приложение в маленькую иконку на вашей панели задач, это удобно при регулярной синхронизации папок.**

> «Хорошо бы кто-нибудь создал нечто подобное для музыкальных файлов.»

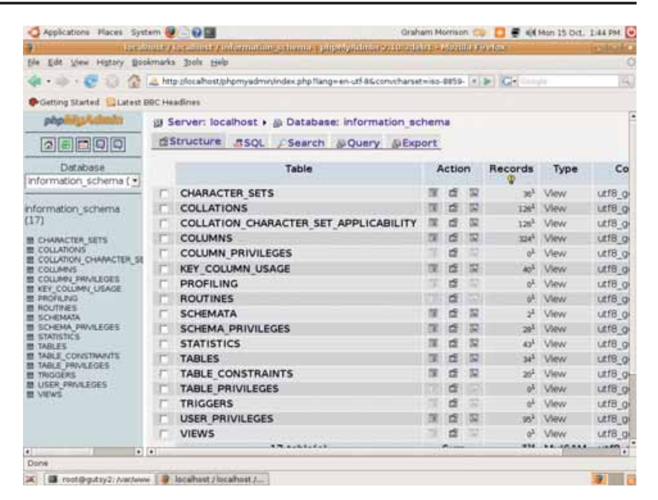
Оболочка MySQL

phpMyAdmin

Версия 2.11.1 **Сайт** www.phpmyadmin.net

phpMyAdmin – одно из тех приложений, которые все используют, но редко отдают им должное. Ненавистникам баз данных оно делает работу приемлемой; тем, кто не в восторге от работы с большими таблицами, оно существенно облегчает задачу. Не берите в голову, что нужно озаботиться его установкой на web-сервер и запуском на PHP: это просто добавит возможность удаленно администрировать базы данных из любого места через Интернет. Название «тонко» намекает на связь с базами данных *MySQL*, и *phpMyAdmin* совместим с версиями 4 и 5 этой почтенной СУБД, хотя разработчики недавно объявили, что прекращают поддержку более старых версий. Как приложение, используемое через браузер, *phpMyAdmin* замечательно. Установка его не сложна, так как почти каждый дистрибутив включает *phpMyAdmin* по умолчанию. Если у вас пока нет работающей базы данных, придется также установить *MySQL*-server.

После установки, *phpMyAdmin* будет доступен через ваш web-браузер по адресу http://localhost/phpadmin. Используйте «root» как имя пользователя и пароль для учетной записи root в *MySQL*. Если имя пользователя и пароль еще не установлены, можно оставить поля пустыми; если установщик предлагает вам установить пароль, используйте вышеупомянутое. Теперь загружайте любую базу данных *MySQL* при помощи выпадающего списка слева. Преимущество *phpMyAdmin* в том, что легко просматривать любую таблицу, какую хотите, и фильтровать результаты по полю и типу данных. Поля можно редактировать на месте, а SQL-запросы писать прямо в окно запросов SQL. Это далеко не все функции *phpMyAdmin* – про него было написано несколько книг; но он легок в использовании и понравится вам больше, чем *MySQL* с командной строкой. Если вы имеете дело с *MySQL*, настоятельно рекомендуем вам установить *phpMyAdmin*: последняя версия значительно переработана, и он сейчас лучше, чем когда-либо.



› **Каким бы сложным ни казался *phpMyAdmin*, работать в нем намного легче, чем с базой данных в его основе.**

> «Удобен для просмотра любых таблиц и фильтрует результат по полю и типу данных.»



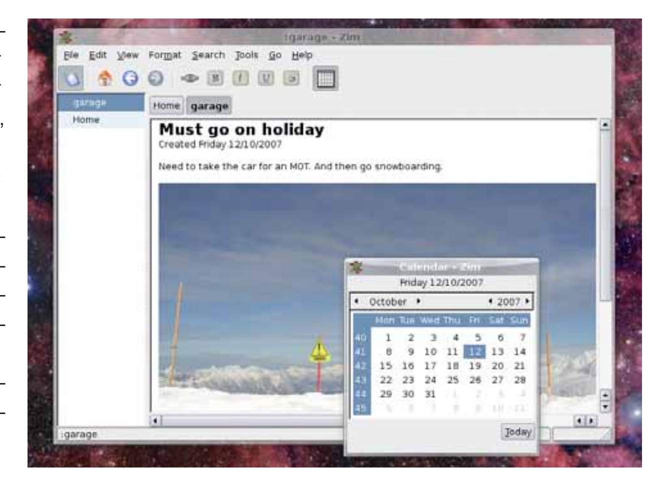
Настольный wiki

Zim

Версия 0.22 **Сайт** http://pardus-larus.student.utwente.nl/~pardus/projects/zim/

Zim был нашим Hottest Pick в LXF81 (Июль 2006). По словам главного разработчика, в *Zim* WYSIWYG означает, что можно видеть и использовать контекстные ссылки внутри документа, как на web-странице. Иначе говоря, *Zim* – это wiki для вашего рабочего стола. Вы можете мигом организовать ссылку на другой документ, выделив слово в редакторе и нажав Ctrl+I: создастся новый документ, названный словом, которое вы выделили. Это немного напоминает *Tomboy* в Gnome – существует даже скрипт для конвертации заметок из *Tomboy* в *Zim*, и предполагается, что он будет использован как утилита для создания записей. Процесс запуска не изменился, хотя мы сочли его несколько путаным: вас спрашивают про хранилище для локальных страниц, и это дезориентирует, поскольку все, что от вас требуется – создать папку в вашем домашнем каталоге. Но мы прощаем *Zim* мелкую проблему в удобстве использования: главное, что редактирование документов работает очень хорошо. Существенные доработки к предыдущим версиям включают поддержку drag-and-drop для внешних файлов (т.е. можно перетащить изображение прямо в окно редактирования), диалоговое окно поиска и замены, функцию импорта экранных снимков, редактор формул и управление шаблонами для новых страниц. Приложение развивается быстро, и скоро последуют новые изменения.

› **Иногда лучше иметь одно приложение для заметок, чем пользоваться методами *Tomboy*. Хотя истории документа все еще нет.**



Управление ресурсами

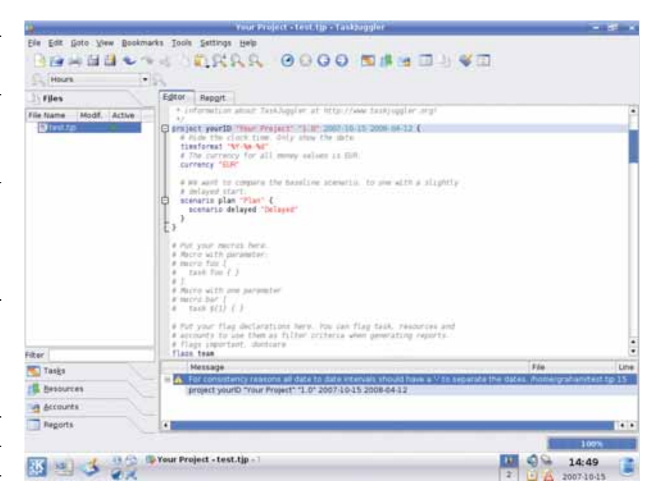
TaskJuggler



Версия 2.4 **Сайт** www.taskjuggler.org

Эта утилита для менеджеров проектов совершенно преобразилась со времени выпуска LXF67. Она выполняет такие важные задачи, как управление ресурсами (со встроенным балансом и проверкой состоятельности) и вывод про-стандартных диаграмм Ганта, которые широко используются среди менеджеров больших проектов для слежения за расходом ресурсов по времени и позволяют перераспределить ресурсы и успеть к сроку. В *TaskJuggler* ресурсы перечислены в текстовом файле, и истые линуксоиды немало порадуются тому, что редактирование этого файла больше похоже на программирование, чем на управление проектом. Ваши данные структурируются, затем *TaskJuggler* просматривает описания примерно так же, как компилятор – код; в итоге этого процесса строятся диаграммы Ганта и заполняются данными другие части программы. Щелчок на любой части данных вернет вас в структуру в исходном текстовом файле. Имеется подсветка синтаксиса, и программа создает впечатление интегрированной среды разработки, превосходной для менеджера проектов. Имеется несколько шаблонов – начинайте с редактирования одного из них, чтобы подстроить его под собственные нужды. Работа с ресурсами при помощи этого приложения становится очень эффективной. LXF

› **На вид больше похоже на IDE, чем на утилиту для управления проектами, и это должно понравиться программистам!**



Также выпущены

Новые и обновленные программы, заслуживающие вашего внимания...

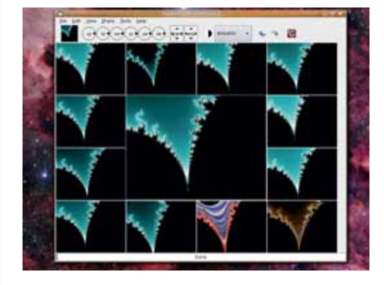
» **KWest 1.1.1** Делитесь вашими интерактивными играми или найдите классику, которой вам не хватает. *KWest* – это лучший из виденных нами интерфейсов. http://kwest.sourceforge.net

» **OSS 4.0 OSS** появилась до ALSA, и она все еще дорабатывается. В этой версии добавлена поддержка некоторых новых аудиоустройств. www.opensound.com

» **Clonezilla 1.0.6-5** Самый близкий эквивалент Norton Ghost для пользователей Linux. Воспроизводит ваш жесткий диск бит за битом. http://clonezilla.sourceforge.net

» **XSensors 0.60** Классические сенсоры рабочего стола добавляют поддержку новых материнских плат и последних процессоров. www.linuxhardware.org

» **Gnofract 4D 3.5** Если вы с 80-х годов не глядели на фрактальные генераторы, теперь самое время попробовать. В этом релизе добавлен компилятор формул. http://gnofract4d.sourceforge.net



› **Сегодня *Gnofract* – один из лучших в Linux способов исследовать красоты фрактальных структур.**

» **Crystal Space 1.2** Этот портативный трехмерный игровой движок очень похож на Ogre; добавлен менеджер копирования и улучшено моделирование земной поверхности. www.crystalspace3d.org

» **xine 1.1.8** Один из двух кандидатов на роль лучшего медиа-проигрывателя в Linux. В этой версии улучшена поддержка DVB-субтитров. www.xinehq.de

» **x11vnc 0.9.3** В этом релизе добавлена возможность кэширования VNC-соединений на клиенте, капитально улучшившая производительность удаленного рабочего стола. www.karlrunge.com/x11vnc/

» **tenshi 0.9.1** Ранее известная как Wasabi, это обновленная версия одной из лучших утилит для управлениями журналами с графическим выводом. http://dev.inversepath.com/trac/tenshi

LXF DVD100

Огромный диск на 8 ГБ с новыми релизами Ubuntu и OpenSUSE!



Отсюда начинается Linux

Майк Сондерс любовно подбирает содержимое диска Linux Format, а также поддерживает сайт www.linuxformat.co.uk.

Отправившись в путешествие по Тропе Памяти, я дошел до *Distro Roundup*, написанного мною для LXF01. Да, более семи лет назад! Интересно наблюдать за тем, какие дистрибутивы выжили, а какие сошли с дистанции – с нами по-прежнему Red Hat, Debian, SUSE и Mandriva, а вот Definite, WinLinux и Caldera OpenLinux отправились в большую коробку из-под ботинок – специально для дисков – на небесах.

В мае 2000 года на нашем диске появился Definite Linux 7.0, ответвление Red Hat, поддерживаемое одной британской компанией. Хотя Definite просуществовал недолго, те, кто стоял за ним, продолжили свою деятельность и запустили интернет-провайдер **UKLinux.net** и магазин дисков **cheeplinux.com**. В то же время у Caldera OpenLinux был шанс на господство на новорожденном рынке настольного Linux для предприятий, однако этот план разлетелся вдребезги, когда Caldera приобрела UnixWare, переименовалась в The SCO Group и с головой ушла в судебные дела.

Однако дистрибутивы могут не только исчезать через пару месяцев существования, они могут и появляться из ниоткуда, и обретать массивную поддержку. Кто бы мог подумать несколько лет назад, что ни один из триумвиров – Mandriva, Red Hat (Fedora) или SUSE – не станет доминирующим дистрибутивом? Что произойдет еще через три года – найдется ли кто-то, кто свергнет Ubuntu с пьедестала?

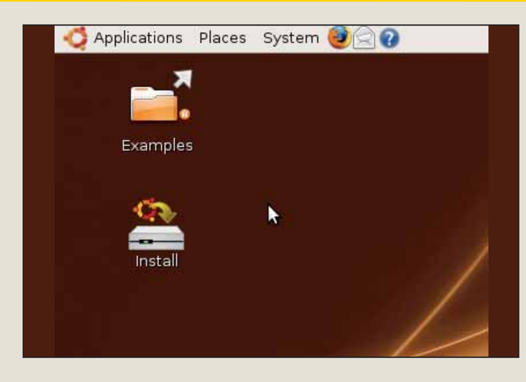
В том-то и прелесть столь динамичной индустрии. В этом выпуске мы представляем вам самые свежие релизы Ubuntu и OpenSUSE на двустороннем DVD. Если вы никогда раньше не пользовались Linux, то Ubuntu может стать превосходной стартовой точкой: поскольку он запускается прямо с диска, вам не надо устанавливать его на жесткий диск. А OpenSUSE – превосходный выбор для более опытных пользователей, в нем такой большой выбор пакетов. Наслаждайтесь!

mike.saunders@futurenet.co.uk

Шаг за шагом: Устанавливаем Ubuntu 7.10



1 Загрузка
Загрузите компьютер с DVD (или образа на CD-R, если у вас AMD64 или вам нужен альтернативный подход) и нажмите Enter в этом меню. Если у вас возникнут проблемы, перезагрузитесь и выберите Safe Graphics.



2 Рабочий стол
Когда появится рабочий стол, можете исследовать Linux. Щелкните на красной кнопке справа вверху, чтобы выйти из системы, или дважды щелкните по значку Install на рабочем столе, чтобы установить себе дистрибутив Ubuntu.



5 Разбиение диска на разделы вручную
Если вы решите разбивать диск на разделы вручную, создайте для Linux корневой раздел (/) ext3 размером не менее 10 ГБ и раздел подкачки для виртуальной памяти размером 512 МБ.



6 Пользователь
Программа установки предложит вам списать настройки Windows, если у вас двойная загрузка, а затем предложит создать имя пользователя и пароль. Не забудьте их!

Дистрибутив Linux

Ubuntu 7.10

Почему Ubuntu стал самым популярным дистрибутивом? Возможно, благодаря сверхпростому процессу установки: несколько минут – и вы в Linux. Или благодаря обширному архиву пакетов, унаследованному от Debian GNU/Linux, дающему доступ к тысячам программ. А возможно, и благодаря сообществу на **www.ubuntuforums.org**, всегда готовому помочь. Тем, кто уже пользуется Ubuntu, все это известно, но если вы только присматриваетесь к Linux, Ubuntu может стать отличной точкой старта. Вот скорострельный перечень новых функций в версии 7.10:

» Эффекты *Compiz-Fusion* добавляют рабочему столу прозрачности и блеска.

» Улучшенное управление дисплеем для смены разрешения и использования нескольких мониторов.

» Новая утилита поиска сканирует документы, историю чатов и т.п.

Естественно, Ubuntu 7.10 также включает самые свежие версии ПО (Gnome 2.20, *OpenOffice.org 2.3.0* и *Firefox 2.0.0.6*). Но! На нашем диске – особая, расширенная версия **LXF** со многими другими пакетами. Мы включили в нее KDE, *Xfce, Mozilla Thunderbird, AbiWord, Gnumeric, KOffice*, инструменты разработки и многое другое, фактически предоставив вам Ubuntu, Kubuntu и Xubuntu в одном-единственном дистрибутиве! Теперь после установки вам есть чем заняться и без нудной загрузки каких-то там дополнительных пакетов.

Ubuntu запускается прямо с нашего DVD, так что вы можете ознакомиться с Linux, не устанавливая его на жесткий диск. Загрузите компьютер с первой стороны диска, и через пару минут вы увидите рабочий стол. Теперь вы можете разобраться с меню Applications вверху слева, попробовать программы и выйти из системы, нажав на красную кнопку вверху справа. Не забудьте, что Ubuntu с DVD будет работать гораздо медленнее, чем с жесткого диска.

Хотите, чтобы Ubuntu прописался у вас? Дважды щелкните по значку Install на рабочем столе и следуйте нижеприведенным инструкциям по установке. Мы рекомендуем 1-ГГц процессор с 256 МБ ОЗУ и 10 ГБ на жестком диске (вы можете переразбить разделы Windows, чтобы изыскать необходимую жилплощадь). Процесс установки очень простой – читай себе указания на экране, но, как и при установке любой операционной системы, сперва сделайте резервные копии данных!

Наша версия Ubuntu прекрасно будет работать на 32-битных и 64-битных x86-64 машинах. Однако если у вас 64-битный компьютер и вы предпочитаете чисто 64-битную версию, мы включили в раздел **Дистрибутивы/Ubuntu** (сторона 1) ISO-образ CD, который вы можете записать на диск и загрузиться с него. Если у вас машина с менее чем 256 МБ ОЗУ, или вам нужен больший контроль над процессом установки, запишите ISO-образ CD 'Alternate' и загружайтесь с него – это обычная 32-битная версия Ubuntu, но с текстовым инсталлятором, экономящим память.

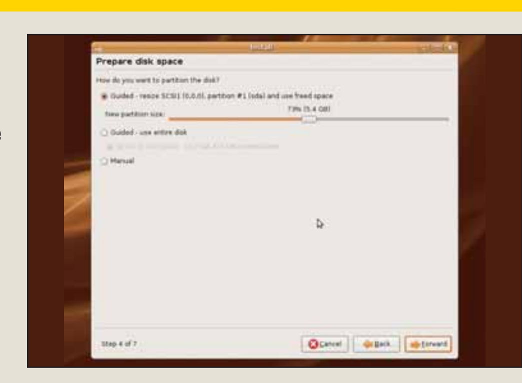
Обратите внимание, что в Ubuntu нет пароля для администратора (root); вместо него для аутентификации вы используете свой пароль обычного пользователя. Если вы хотите начать root-сессию, откройте терминал из меню Applications и введите *sudo bash*. »



› Наша особая версия Ubuntu «супер-окрошка» включает KDE и изобилие инструментов разработки!



3 Язык
Запустится программа установки; щелкните Forward для выбора часового пояса по карте (чтобы увеличить карту, щелкните по ней). Затем снова нажмите Forward и выберите раскладку клавиатуры.



4 Разделы
Теперь отведите место на жестком диске под Linux. Чтобы освободить место, можете переразбить уже существующие разделы (рекомендуется отвести 10 ГБ), выделить для Linux весь жесткий диск или разбить его на разделы вручную.



7 Подтверждение
Тут программа установки представит краткий отчет о своих действиях – нажмите Install, чтобы она начала работу. (Если вам не нужен загрузчик, нажмите Advanced).



8 Копирование
Программа установки начнет копировать Ubuntu на ваш жесткий диск. После завершения этой операции она перезагрузит компьютер – извлеките DVD и наслаждайтесь новой установкой Linux!

Дистрибутив Linux

OpenSUSE 10.3

OpenSUSE – один из самых зрелых и уважаемых дистрибутивов Linux: первая версия SUSE появилась в 1994 году, а ныне OpenSUSE – отличный выбор для опытных пользователей Linux: в нем имеются рабочий стол и ПО для разработки и для сервера со впечатляющим набором инструментов (Для новичков это тоже отличный дистрибутив, но он не работает в режиме Live, поэтому начинающим мы рекомендуем Ubuntu). Новинки версии 10.3:

» **KDE 4 preview:** рабочий стол по умолчанию – KDE 3.5.7, с несколькими кусочками из KDE 4 (игры и клиент доступа к удаленному рабочему столу). Но вы можете установить полную предварительную версию KDE 4 – только помните, что на данной стадии в ней могут быть ошибки!

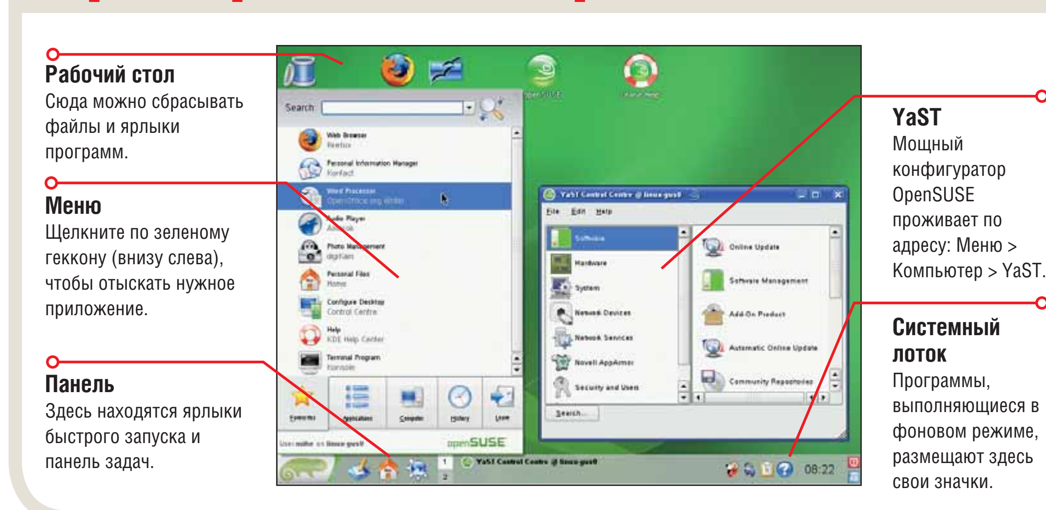
» **Красочная новая тема:** она сохраняет традиционный для SUSE зеленый цвет, но с завихрениями в стиле Apple. Пересмотрено все – от загрузочного меню до экрана приглашения.

» ***Compiz:*** Отражения, вращающиеся кубы и эффекты увеличения – все, чтобы придать вашему рабочему столу пикантности. Для улучшения производительности потребуется хорошая видеокарта ATI или Nvidia (встроенные чипы Intel тоже работают, но гораздо медленнее).

» **Gnome 2.20:** Хотя пользователи OpenSUSE в основном отдают предпочтение KDE, Gnome, тем не менее, тоже популярен, и теперь пользователи основанного на *Gtk 2* рабочего стола получили идеально сюда подходящую версию инструмента настройки *YaST*.

» **Ускорение загрузки:** Исторически OpenSUSE всегда был одним из наиболее медленно загружающихся дистрибутивов, но разработчики улучшили время запуска, в некоторых случаях сократив его вдвое!

Изучаем рабочий стол OpenSUSE



Рабочий стол
Сюда можно сбрасывать файлы и ярлыки программ.

Меню
Щелкните по зеленому геккону (внизу слева), чтобы отыскать нужное приложение.

Панель
Здесь находятся ярлыки быстрого запуска и панель задач.

YaST
Мощный конфигуратор OpenSUSE проживает по адресу: Меню > Компьютер > YaST.

Системный лоток
Программы, выполняющиеся в фоновом режиме, размещают здесь свои значки.

» **Виртуализация:** *Xen 3.1*, *QEMU* и *VirtualBox* позволяют запустить другие ОС в гостевом режиме.

Также проведена обычная работа по обновлению программ и небольшим усовершенствованиям, и все построено на основе ядра 2.6.22.5, *glibc 2.6.1*, *GCC 4.2* и *X.org 7.2*. На рабочем столе вы найдете *OpenOffice.org 2.3*, *Seamonkey 1.1.4*, *KOffice 1.6.3*, *AbiWord 2.4.6* и *Window Maker 0.92*, а пользователи серверов ухватятся за *Apache 2.2.4*, *Sendmail 8.14.1*, *Postfix 2.4.5*, *MySQL 5.0.45* и *PostgreSQL 8.2.4*. Инструменты разработки включают Python 2.5.1, Perl 5.8.8, Ruby 1.8.6 и PHP 5.2.4. На DVD более 2 800 пакетов!

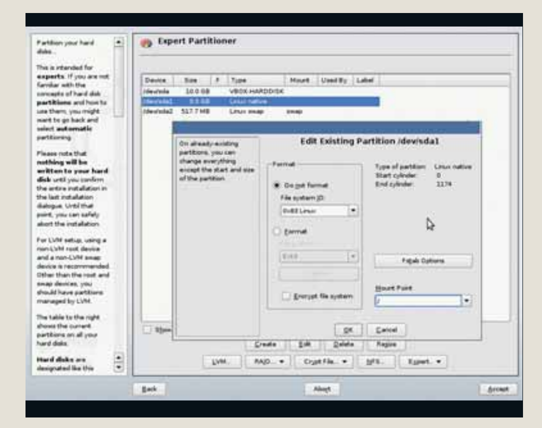
» **Системные требования:** для настольного ПК рекомендуем процессор 1 ГГц (32-битный или x86-64) с 512 МБ ОЗУ и 10 ГБ свободного места на жестком диске. Если вы решили использовать OpenSUSE в качестве сервера командной строки, вам хватит машины с 300 МГц, 256 МБ ОЗУ и 2 ГБ свободного места на диске. Для установки дистрибутива, загрузите компьютер со второй стороны нашего DVD.

OpenSUSE загрузит свою программу установки; следуйте инструкциям: примерно через час (в зависимости от скорости вашего компьютера) у вас будет установлен новый OpenSUSE 10.3. Чтобы настроить компьютер после установки, используйте *YaST2*, утилиту настройки Novell, она охватывает все аспекты вашей системы. Если при установке дистрибутива возникнут проблемы, ссылки на документацию имеются на http://ru.opensuse.org, и вы сможете попросить помощи у других пользователей OpenSUSE на www.linuxforum.ru. Развитесь!

Шаг за шагом: Устанавливаем OpenSUSE 10.3



1 Загрузка
Загрузите компьютер со стороны 2 LXF DVD, и вы увидите это меню. Выберите Installation и нажмите Enter. Появится программа установки; читайте и следуйте инструкциям.



2 Разделы
Нажмите Partitioning на экране Settings, если вы не собираетесь отдать весь жесткий диск под Linux. Вам потребуется корневой раздел (/) ext3 размером не менее 5 ГБ и раздел подкачки (swap) размером 512 МБ для виртуальной памяти.

Дополнительное ПО

Новые релизы

Также на **DVD** этого месяца (Сторона 1) вас ожидает солидная подборка лучших релизов мира Open Source. Начнем с раздела **Рабочий стол**: *Osmo* – личный органайзер – объединил календарь, менеджер задач и адресную книгу. Если, подобно нам, вы раньше использовали эти функции на КПК и цифровых записных книжках и пытались найти их эквивалент для ПК, то *Osmo* – то, что надо. В отличие от мощных *Evolution* или *Kontact*, *Osmo* не претендует на звание всеобъемлющего решения для бизнес-пользователей: вместо этого он придерживается самых основ и очень прост в работе. Хотя *Osmo* покамест на находится ранней стадии развития, он уже является многообещающим и удобным инструментом, достойным уголка на рабочем столе.

Есть и новые программы для технарей: *Avogadro*, *Happydigger* и *Qosmic*. Первый – молекулярный редактор, он умеет соединять атомы в молекулы химических веществ. Используя OpenGL, *Avogadro* показывает молекулы во всей их трехмерной красе; он построен на библиотеке *OpenBabel*, позволяющей импортировать всевозможные форматы химических файлов. На наш **DVD** мы включили самый свежий исходный код – чтобы собрать его, вам понадобится *Qt*, *Eigen* и вышеупомянутый *OpenBabel* (см. папку **dependencies**).

Happydigger – утилита каталогизации археологических находок (как интерфейс базы данных *SQLite*). Вы можете вводить записи в форму, а *Happydigger* будет их отслеживать. Программа позволяет задавать материалы, исторические периоды и комментарии к вашим находкам, и даже снабжает их фотографиями.

Qosmic тоже можно отнести к категории программ для специалистов, но он повеселит даже тех, кто ничего не знает о фрактальных изображениях. Эта программа позволяет создавать пышные эффекты пламени, изменяя разные статистические данные и настройки. Для его запуска вам понадобятся установленные *Qt 4* и пакеты разработки; найдите в менеджере пакетов *qt4-dev*, или *libqt-dev*, или нечто подобное. Затем скопируйте *qosmic-1.1.2.tar.bz2* с DVD в домашнюю директорию, откройте окно терминала и введите:

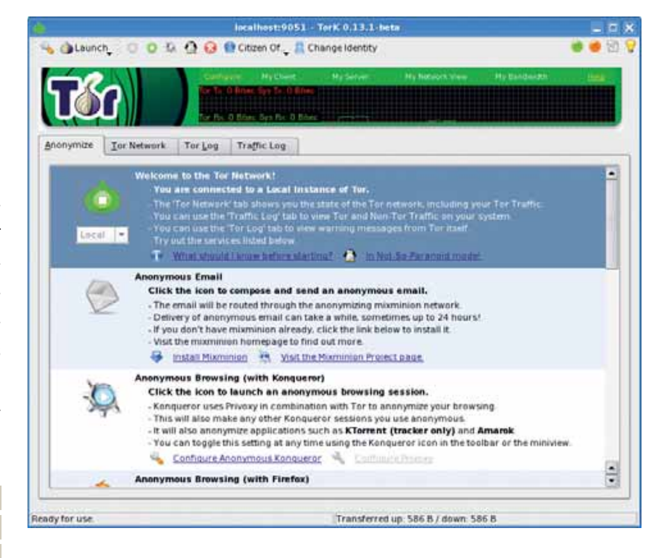
```
tar xfvj qosmic-1.1.2.tar.bz2
cd qosmic
./build.sh
```

Если все пойдет хорошо, и нужные пакеты разработки у вас есть, *Qosmic* скомпилируется, и вы наконец-то сможете запустить программу, введя **./qosmic**.

Интернет и OOo

В разделе **Интернет** у нас есть *Quack*, приложение Gnome (хотя оно работает и в KDE), позволяющее размещать сообщения под учетной записью Jaiku. Вы спрашиваете, что такое Jaiku? Это – микроблоговый сервис, во многом похожий на Twitter, для размещения коротких заметок на сайте. В противоположность «большим» блог-сервисам, специализирующимся на длинных сообщениях и ссылках на другие сайты, Jaiku предназначен для быстрых обновлений и обмена сообщениями с друзьями (общение через сеть). Jaiku заинтересовал Google, так что можно рассчитывать на рост популярности сервиса – пора резервировать местечко в экипаже *Quack*!

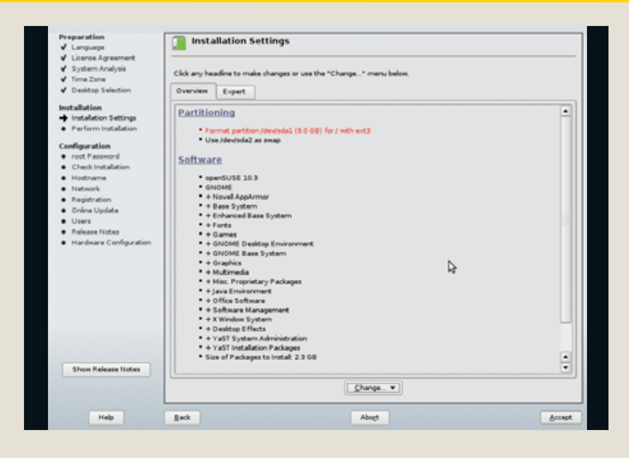
Если вам хочется скрыть, чем вы занимаетесь в Интернет, попробуйте *TorK*, интерфейс KDE для сервиса анонимности Tor. Tor направляет ваш трафик в Интернете через случайно выбранные компьютеры; в конце концов для удаленного сервера становится невозможным отследить, откуда пришел изначальный запрос. *TorK* добавляет сервису глянец KDE: стоит взглянуть.



› Заметите следы в Интернет с помощью сервиса анонимности Tor и *TorK* от Electronic Frontier Foundation.

Наконец, на DVD прошлого месяца был *OpenOffice.org 2.3*, а сейчас мы рады представить вам сборку Pro от компании «Инфра-ресурс». Если вы пропустили LXF99, вот краткие инструкции. Мы включили на диск два архива: один с RPM-пакетами (Fedora, Mandriva, SUSE), а другой – с *.deb* (Ubuntu, Debian, Mepis). Удалите все существующие установки *OpenOffice.org*, распакуйте соответствующие архивы *.tar.bz2* в файловом менеджере и установите эти пакеты через менеджер. После установки можете запускать *OOo*, введя в терминале (или создав ярлык на рабочем столе):

```
/opt/openoffice.org2.3/program/soffice
```



3 Настройка

Экран Настройки [Settings] поможет вам подобрать нужные опции перед копированием файлов. Затем добавьте учетные записи, выполните настройку – и вы в системе!

И наконец...

LXF100 – это, несомненно, юбилей, ну а какой же юбилей без подарков? Поэтому на диске этого номера вы найдете подшивку всех выпусков LXF (исключая текущий), которые когда-либо выходили на русском языке. Неважно, читаете ли вы LXF с пилотного 70-го номера или впервые оформили подписку на 2008 год – теперь у вас на руках будут все материалы, которые команда *Linux Format* подготовила за два с лишним года! Вы научитесь программировать на Perl и Python, Java и Mono, узнаете *Qt*, *GTK+*, *wxWidgets*, освоите *LaTeX*, *Maxima* и *Blender*, и, надеемся, найдете для себя еще много чего интересного. Не забывайте, что мы приглашаем всех желающих принять участие в создании wiki-архива журнала *Linux Format*: напишите нам на info@linuxformat.ru, чтобы узнать все подробности. Наслаждайтесь чтением! LXF

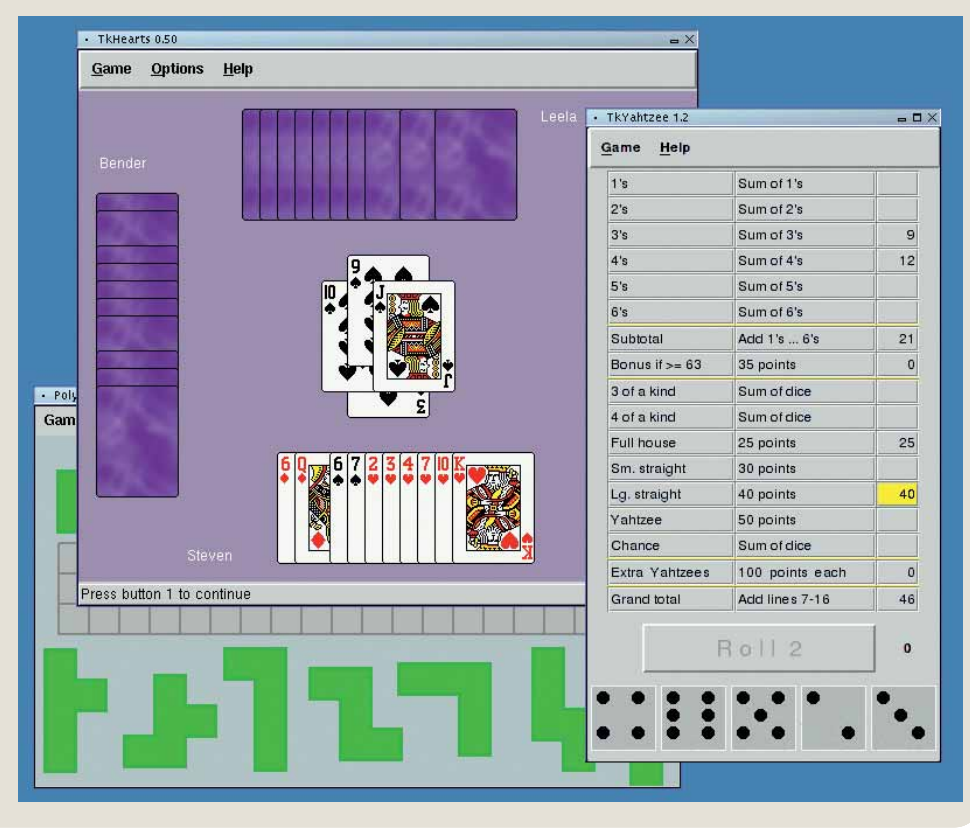


› *Linux Format* – читают все пользователи Linux!

Содержание DVD



› Нужно создать симпатичные наклейки или визитные карточки? Поможет *gLabels*.



***› TkGames*: Червы, Паук, Виселица, Yahtzee и Солитер – обеденный перерыв промчится незаметно!**

DVD

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ DVD-ДИСК !

Информация о диске

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;
имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;
имя_программы-1.0.1.tar.gz – обычно это исходный код;
имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;
имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;
имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

LINUX FORMAT 8 GIGABYTE DVD



openSUSE™

10.3

Универсальный дистрибутив Linux:
KDE 3.5.7, GNOME 2.20, OpenOffice.org 2.3 и 2800 других пакетов

Январь 2008

LXFDVD100-101



Эксклюзивная улучшенная версия самого популярного дистрибутива !

ubuntu
Последний релиз – 7.10!
Включает GNOME, KDE и Xfce – на выбор
Эффекты рабочего стола Compiz Fusion
1500 пакетов на любой вкус!

Бонус! Подшивка LXF за 2005-2007 год!



Содержание LINUX FORMAT DVD Январь 2008

Сторона 1

Сторона 1

Рабочий стол
Avogardo - кроссплатформенный редактор для моделирования молекул.
Blender - профессиональный 3d-редактор.
Gnome - рабочее окружение.
Happydigger - каталогизатор находок.
Muine - инновационный музыкальный проигрыватель.
Osmo - удобный персональный органайзер.
Qosmic - редактор фракталов.
Viking - программа для работы с GPS.

Разработка
Arduino_IDE - среда разработки для микроконтроллеров.
Mono - реализация .NET.
Net-IRC - Perl-модуль для работы с IRC.
Nixstaller - утилита для создания инсталляторов.

Дистрибутивы
Ubuntu - полнофункциональный дистрибутив Linux.

Игры
Bomb attack - клон Bomberman.
Lost_Labyrinth - ролевая игра.
TkGames - сборник мини-игр.
Typhoon 2001 - клон Tempest.

Справка
Ответы на часто задаваемые вопросы.
RUTE - книга по администрированию Linux.

Hot Picks
Battle Tanks - забавные танковые сражения.
Bos Wars - футуристическая стратегия.
Gnome Inform 7 - среда разработки для Inform 7.
Motion - детектор движения.
phpMyAdmin - комплект для простого управления MySQL.
ProjectM - программа визуализации для XMMS.
Synkron - утилита для синхронизации директорий.
TaskJuggler - программа для управления проектами.
Yakuake - эмулятор терминала для KDE.
Zim - WYSIWYG-редактор в wiki-стиле.

Интернет
DC# - клиент для сетей Direct Connect.
Quack - интерфейс к Jaiku.com
TorK - центр управления Tor для среды KDE.
wxCam - программа для работы с веб-камерами в Linux.

Офис
gLabels - программа для создания визиток и этикеток.
OpenOffice.org - свободный офисный пакет.

Система
cryptsetup-luks - замена cryptsetup.
Firebird - реляционная база данных.
FlameRobin - инструмент для администрирования Firebird.
GParted - редактор разделов.
StartUp Manger (SUM) - утилита для изменения настроек Grub/Grub2 и Usplash.

Поставляется вместе с журналом LINUXFORMAT номер 1 (100-101) Январь 2008

Сторона 2

OpenSUSE 10.3

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Фортмедиа», Россия, Санкт-Петербург, 196006 ул. Цветочная д. 7, тел. +7 (812) 388-8290. Лицензия МПТР России ВАФ № 77-225




Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF


ORACLE
AMD
intel
ERICSSON


Sun
microsystems



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 6000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Родион Водейко, Елена Толстякова, Иван Мищенко
Переводчики
Илья Аввакумов, Александр Бикмеев, Светлана Кривошеина, Александр Казанцев, Алексей Опарин, Валентин Развозжаев, Татьяна Цыганова, Юлия Шабунио
Редактор диска
Александр Кузьменков
Допечатная подготовка
Мария Пучкова, Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 921 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____

Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Редактор новостей Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Редактор обзоров
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Литературный редактор
Мэтт Нейлон (Matt Nailon) mnailon@futurenet.co.uk

Подготовка материалов
Ладислав Боднар [Ladislav Bodnar], Нейл Ботвик [Neil Bothwick], д-р Крис Браун [Dr. Chris Brown], Энди Ченнел [Andy Channelle], Энди Хадсон [Andy Hudson], Джек Найт [Jack Night], Дэниэл Джеймс [Daniel James], Джо Касселс [Joe Cassels], Евгений Балдин, Андрей Боровский, Андрей Паскаль, Галина Пожарина, Александр Поносов, Андрей Прахов, Петр Семилетов, Александр Супрунов, Алексей Федорчук, Алексей Шипунов, Илья Шпаньков

Художественные ассистенты: Дэмиан МакГи [Damian McGee], Джон МакЭллистер [John McAllister], Нейл Фристоун [Neil Freestone], Клэр Герри [Claire Gerrey], Ник Кокс [Nick Cox], Стейси Блек [Stacey Black], Роберт Хейл [Robert Hale]

Иллюстрации: Пол Блечфорд [Paul Blachford], Крис Винн [Chris Winn], iStock Photo

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
СПб (редакция): ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Email: info@linuxformat.ru, Web: www.linuxformat.ru
Представительство в Москве: ул. Энергетическая, д.14, корпус 5, стр. 1
По вопросам сотрудничества, партнерства, оптовых закупок: +7 (495) 799-18-63
Дирекция московского офиса: +7 (495) 136-88-45. E-mail: moscow@linuxcenter.ru



Вторая жизнь старого ПК

Старый Pentium III пылится в углу? Недостаток вычислительных мощностей и памяти – не проблема, Linux выжмет из него все до последней капли!

Текстовые процессоры на базе Web

Является ли *OpenOffice.org* конкурентом для *Firefox*? Мы сравнили лучшее, что может дать Сеть.

GIMP для фотографа

Версия 2.4 наконец-то созрела для редактирования на профессиональном уровне.

X3: Elite 2008 года?

X2+1 добралась и до Linux, и мы погрузились в игру с головой

Содержание последующих выпусков может меняться без уведомления.

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА В ЛИНУКСЦЕНТРЕ

Сколько стоит подписка?

Подписка на журнал «Linux Format» 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь) стоит 1800 рублей
Подписка на журнал «Linux Format» 6 номеров (июль, август, сентябрь, октябрь, ноябрь, декабрь 2008 года) стоит 900 рублей

Как оформить подписку?

Чтобы оформить подписку на журнал «Linux Format», необходимо зарегистрироваться в интернет-магазине Linuxcenter.Ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар «Подписка на журнал «Linux Format» 12 номеров 2008 года», или товар «Подписка на журнал «Linux Format» второе полугодие 2008 года», получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Как оплатить подписку?

- по выставленному счету (для юридических лиц)
- по квитанции в любом отделении Сбербанка

Плюсы подписки

- подписка дешевле!
- гарантированное получение нового номера журнала!

ПОДПИСКА - 2008!

ПОДПИСКА ПО КАТАЛОГАМ РФ

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882

Каталог «ПРЕССА РОССИИ» – подписной индекс 87974

Ф. СП-1

Министерство связи РФ

АБОНЕМЕНТ НА ЖУРНАЛ

Linux Format

индекс издания

количество комплектов

НА 2008 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА ______ почтовый индекс ______ адрес доставки

КОМУ ______ фамилия, инициалы

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ | МЕСТО | ЛИТЕР

НА ЖУРНАЛ ______ индекс издания

Linux Format

наименование издания

СТОИМОСТЬ	подписки	руб.	коп.	количество комплектов
	переадресации	руб.	коп.	

НА 2008 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА ______ почтовый индекс ______ адрес доставки

КОМУ ______ фамилия, инициалы



ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА ПО КАТАЛОГАМ СНГ И БЛИЖНЕГО ЗАРУБЕЖЬЯ

Каталог «Российская Пресса» – совместный проект Государственного предприятия «Казпочта», Агентства «Книга-Сервис» и АРЗИ.

Блок изданий АРЗИ в национальных Каталогах Украины и Беларуси. В Азербайджане, Армении, Грузии, Киргизии, Узбекистане и Молдове - по изданиям, включенным в Объединенный каталог, распространяемые через АРЗИ.

Азербайджан

- по Объединенному каталогу российских изданий через Предприятие по распространению печати «Гасид»(370102, г. Баку, ул. Джавадхана, 21);

Армения

- по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г.Ереван, пл.Сасунци Давида, д.2) и ЗАО «Контакт-Мамул» (375002, Г.Ереван, ул.Сарьяна, 22);

Белоруссия

- по Каталогу изданий стран СНГ через РГО «Белпочта» (220050, г.Минск, пр-т Ф.Скорины, 10);

Грузия

- по списку номенклатуры «АРЗИ»» через АО «Сакпресса» (380019, г.Тбилиси, ул.Хошараульская, 29) и АО «Мацне» (380060, г.Тбилиси, пр-т Гамсахурдия, 42);

Казахстан

- по Каталогу «Российская Пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»;

Молдавия

- по каталогу через ГП «Пошта Молдавей» (МД-2012, г.Кишинев, бул.Штефан чел Маре, 134);
- по списку через ГУП «Почта Приднестровья» (MD-3300, г.Тирасполь, ул.Ленина, 17);
- по прайс-листу через ООО Агентство «Editil Periodice» (2012, г.Кишинев, бул. Штефан чел Маре, 134).

Узбекистан

- по Каталогу «Davriy nashrlar» российские издания через Агентство по распространению печати «Davriy nashrlar» (7000029, Ташкент, пл.Мустакиллик, 5/3, офис 33);

Украина

- Киевский главпочтамт.
- Подписное агентство «KSS» Телефон/факс (044)270-62-20, 270-62-22

ПОДПИСКА НА LINUX FORMAT

Агентство “Centerpress”

Сколько стоит подписка?
Подписка на журнал "Linux Format" 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь 2008 года) стоит 1800 рублей.

Как оформить подписку?
Чтобы оформить подписку на журнал "Linux Format", необходимо зарегистрироваться в интернет-агентстве Centerpress.ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар «Подписка на журнал "Linux Format" на 2008 год 12 номеров (01-12/2008)», получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Агентство “Centerpress”: www.centerpress.ru

Все Плюсы подписки!
- Подписка дешевле!
- Гарантированное получение журнала!

По каталогам РФ

Каталог агентства “РОСПЕЧАТЬ” - подписной индекс

20882

Каталог “ПРЕССА РОССИИ” - подписной индекс

87974



АЛЬТЕРНАТИВНЫЕ АГЕНТСТВА РФ

Агентство «Интер-Почта» (495) 500-00-60, курьерская доставка по Москве.

Агентство «Вся Пресса» (495) 787-34-47

Агентство «УралПресс»

- Екатеринбург, Березовский, В. Пышма, Первоуральск тел. (343) 375-80-71, 375-84-93, 375-84-39, факс 375-62-74, info@ural-press.ru
- Нижний Тагил тел. (3435) 411448, 417709, ntagil@ural-press.ru
- Челябинск тел. (351) 262-90-03, 262-90-05, pochta@chel.surnet.ru
- Пермь тел. (3422) 60-24-40, 60-22-95, 60-35-42, parma-press@permonline.ru